

**PCT**WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau

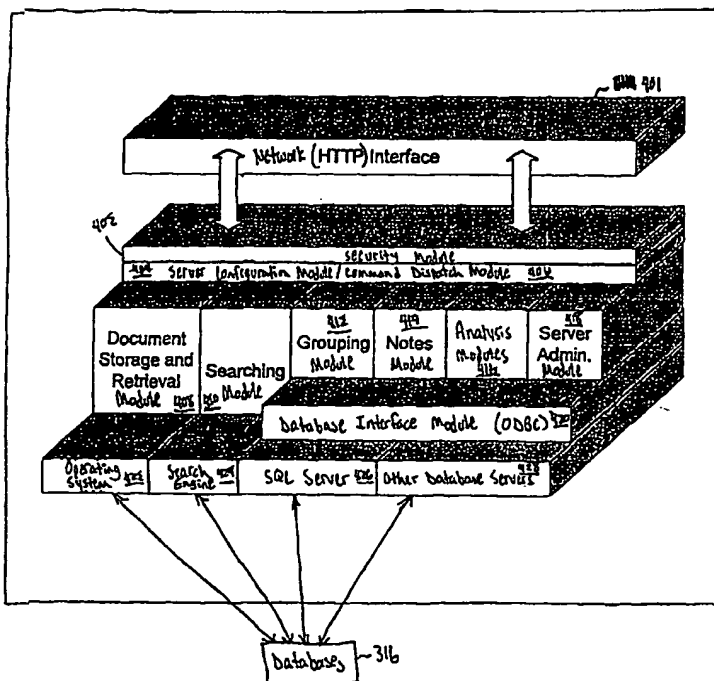
## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> : <b>G06F 17/30, 17/60</b>		A1	(11) International Publication Number: <b>WO 00/11575</b>
			(43) International Publication Date: 2 March 2000 (02.03.00)
(21) International Application Number: PCT/US99/19050			Plateau Drive, Belmont, CA 94002 (US). BRANNON, Donald; 204 Washington Street #4, Santa Clara, CA 95050 (US). SCHNITZ, Matthew; 2558 Mardell Way, Mountain View, CA 94043 (US).
(22) International Filing Date: 23 August 1999 (23.08.99)			
(30) Priority Data: 09/138,368 21 August 1998 (21.08.98) US			
(71) Applicant: AURIGIN SYSTEMS, INC. [US/US]; 1975 Landings Drive, Mountain View, CA 94043-0801 (US).			
(72) Inventors: RIVETTE, Kevin, G.; 2165 Waverley Street, Palo Alto, CA 94303 (US). RAPPAPORT, Irving, S.; 1500 Edgewood Drive, Palo Alto, CA 94303 (US). HOHMANN, Luke; 306 Windmill Park Lane, Mountain View, CA 94043 (US). PUGLIA, David; 17429 East Vineland Avenue, Los Gatos, CA 95030 (US). GORETSKY, David; 272 Waverly Street, Sunnyvale, CA 94086 (US). JACKSON, Adam; 1063 Morse Avenue #7-107, Sunnyvale, CA 94089 (US). RABB, Charles, Jr.; 730 East Evelyn #638, Sunnyvale, CA 94086 (US). SMITH, David, W.; 3 Morning Sun Court, Mountain View, CA 94043 (US). PARK, Brian; 4029 Park Boulevard, Palo Alto, CA 94306 (US). THORNTHWAITE, Warren; 147 Hedge Road, Menlo Park, CA 94025 (US). NAVARRETE, Jorge, A.; 160 Hedge Road, Menlo Park, CA 94025 (US). MULLER, Robert, J.; 36 Whitney Street, San Francisco, CA 94131 (US). ALCABES, Harvey; 3305			
			(74) Agents: LEE, Michael, Q. et al.; Sterne, Kessler, Goldstein & Fox P.L.L.C., Suite 600, 1100 New York Avenue, N.W., Washington, DC 20005-3934 (US).
			(81) Designated States: AU, CA, JP, KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
			Published With international search report.

(54) Title: SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR MANAGING AND ANALYZING INTELLECTUAL PROPERTY (IP) RELATED TRANSACTIONS

## (57) Abstract

A system, method, and computer program product to track, analyze, and report on information related to intellectual property (IP) transactions, including license and related agreements, includes an enterprise server (214), network (212), network clients (206A-C), and databases (216), as well as a web server (210) and web clients (204A-B). The enterprise server (214) includes a network interface (301), a security module (302), a database interface module (320), and various user interface modules such as an Administrator module (318), document storage and retrieval module (308), and Analysis modules (316).



BEST AVAILABLE COPY

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

## **System, Method, and Computer Program Product for Managing and Analyzing Intellectual Property (IP) Related Transactions**

### ***Background of the Invention***

#### **5      *Field of the Invention***

The present invention is generally related to tools for data processing, and more particularly related to tools for patent-centric and group-oriented data processing. The tools include modules to track and process IP related transactions, such as license agreements.

#### **10     *Related Art***

Patents are becoming more and more important to a business's success, especially in today's global economy. Patents can be viewed as a new type of currency in this global economy because they grant the holder with a right to exclude others from making, using, or selling the patented technology. In some  
15 industries, product turnover is fairly rapid. However, core technology, product features, and markets change at a much slower rate. Accordingly, even in fast-moving industries, patents which cover core technology are very valuable at protecting a company's research and development investment for an extended period of time.

20 Patents are also valuable as revenue generators. In 1993, for example, the revenue generated from patents by U.S. companies was over \$60 billion. Fred Warshofsky, *The Patent Wars*, John Wiley & Sons, Inc., New York, 1994. These patent revenue dollars are rising each year.

25 Patents are further valuable because they collectively represent a vast technological database. Much of this database is only available as issued patents (i.e., it is not released in any other form). According to Larry Kahaner's book,

*Competitive Intelligence*, Simon & Schuster, 1996, "More than 75 percent of the information contained in U.S. patents is never released anywhere else."

5 More and more corporations are recognizing the value of patents. The number of patents applied for and issued to U.S. companies is increasing every year, especially in fast moving industries such as computer software and biotechnology. Many international companies have also recognized the value of patents. In fact, foreign companies regularly rank among the leaders in issued U.S. patents.

10 Yet, for all the heightened awareness being paid to patents in some quarters, patents remain one of the most underutilized assets in a company's portfolio. This is due, at least in significant part, to the fact that patent analysis, whether for purposes of licensing, infringement, enforcement, freedom to operate, technical research, product development, etc., is a very difficult, tedious, time consuming, and expensive task, particularly when performed with paper copies of patents.

15 Software providers have been slow in developing software tools for aiding in the patent analysis process. As a result, there are few automated tools for patent analysis currently available. There are software tools available for managing corporate patent prosecution and payment of maintenance fees, such as products from Master Data Corporation. The patent analysis capabilities of these tools are limited. These tools, for example, cannot be used to facilitate the analysis and development of business strategies to increase corporate shareholder value through the strategic and tactical use of patents.

20 A number of patent searching tools are available, such as the United States Patent and Trademark Office (USPTO) Automated Patent System (APS), and the on-line search services offered by Lexis and Westlaw. Other providers of patent information and patent search tools include Derwent, MicroPatent, Questel, Corporate Intelligence, STN, IFI/Plenum, The Shadow Patent Office (EDS), IBM, and CAS. These tools are not analysis tools. Instead, they are search



tools. These tools enable a user to identify patents that satisfy a specified key word search criteria. In essence, these tools provide the user with the ability to possibly find "the needle-in-the-haystack." However, these tools have limited, if any, automated functions to aid a user in analyzing the patents, whether the  
5 company's own patents or those of competitors, for the purpose of making tactical and strategic business decisions based on the patents.

SmartPatents Inc. (SPI) of Mountain View, CA, provides electronic tools for analyzing patents. These tools, collectively called the SmartPatent Workbench, are very useful for analyzing patents. With the SmartPatent  
10 Workbench, a user can view the text and image of a patent, conduct text searches in the patent, copy and paste portions of the patent to other documents, build a case of patents, annotate the case and the patents in the case, import and export patents and cases, etc. The SmartPatent Workbench is commercially available from SPI, and is described in a number of publicly available documents, such as  
15 U.S. Patent No. 5,623,679 and U.S. Patent No. 5,623,681.

For the most part, existing patent-related tools can process only the information contained in patents. (It is noted, however, that the SmartPatent Workbench has functions to annotate patents with any information, whether or not patent related, and has additional functions to search within annotations.)  
20 These tools do not have functions for correlating, analyzing, and otherwise processing patent-related information with non-patent related information, including but not limited to corporate operational data, financial information, production information, human resources information, and other types of corporate information. Such non-patent information is critically important when  
25 evaluating the full strategic and tactical value and applicability of any given patent, or developing a corporate patent business strategy for gaining competitive advantage and increasing shareholder value based on patents.

### ***Summary of the Invention***

Briefly stated, the present invention includes a system, method, and computer program product to track, analyze, and report on information related to intellectual property (IP) transactions, including license and related agreements.

5 Further features and advantages of the invention, as well as the structure and operation of various embodiments of the invention, are described in detail below with reference to the accompanying drawings. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

10

### ***Brief Description of the Figures***

The present invention will be described with reference to the accompanying drawings, wherein:

FIG. 1 illustrates the document-centric and patent-centric operation of the present invention;

15

FIG. 2 is a block diagram of a system according to a preferred embodiment of the present invention;

FIG. 3 is a block diagram of an enterprise server according to a preferred embodiment of the present invention;

20 FIG. 4 is a block diagram of the databases of the present invention;

FIG. 5 is a block diagram of a network client (and potentially a web client) according to an embodiment of the invention;

FIG. 6 is a block diagram of the analysis modules which form a part of the enterprise server of FIG. 3;

FIG. 7 is a block diagram of a computer useful for implementing components of the invention;

FIG. 8A illustrates the orientation of FIGS. 8B-8M relative to one another;

5           FIGS. 8B-8M illustrates the tables and attributes in the databases of FIG. 4 according to an embodiment of the invention;

FIG. 9 represents an example console screen shot;

FIG. 10 is another example console screen shot;

FIG. 11A illustrates a block diagram of a licensing module;

10           FIG. 11B illustrates a more detailed block diagram of the licensing module according to an embodiment of the invention;

FIG. 12 illustrates a standalone configuration of the licensing module;

FIG. 13 illustrates an integrated configuration of the licensing module;

15           FIG. 14 illustrates an example screen shot of an object display window generated by a licensing module according to an embodiment of the present invention;

FIG. 15 illustrates an actor hierarchy according to an embodiment of the licensing module;

20           FIG. 16 is a conceptional diagram of databases used by the licensing module according to an embodiment of the present invention;

FIG. 17A is a block diagram of the functional modules in the licensing module according to an embodiment of the invention;

FIG. 17B is a block diagram of a licensing administration module according to an embodiment of the invention;

25           FIGS. 18, 20, 21, 23-29, 31, 36, 41, 45, 46, 48, 49, 58, 60-63, 73-80, 82-86, 87B, 91-94, 100-107, 109, and 118-124 are diagrams of use cases representing operational functions of the licensing module according to an embodiment of the invention;

FIG. 19 is an example screen shot of a log-in window according to an embodiment of the invention;

FIG. 22 illustrates an example screen shot of a contact view according to an embodiment of the invention;

5           FIG. 30 illustrates an example screen shot of an asset view according to an embodiment of the invention;

FIG. 32 illustrates the use of pull down menus to create new objects according to an embodiment of the invention;

10           FIGS. 33-35 illustrate example screen shots of dialogs for entering new patent assets according to an embodiment of the invention;

FIGS. 37-40 illustrate example screen shots of dialogs for entering new trademark assets according to an embodiment of the invention;

FIGS. 42-44 illustrate example screen shots of dialogs related to entering new copyright assets;

15           FIG. 47 illustrates an example screen shot for entering a new know how asset according to an embodiment of the invention;

FIGS. 50-57 and 59 illustrate example screen shots of dialogs and windows associated with asset packages;

20           FIGS. 64A, 64B, 65, and 66 illustrate example screen shots of dialogs related to a find asset tool;

FIG. 67 illustrates the manner in which a pull down menu can be used to invoke a find tool;

FIGS. 68-72 and 81 illustrate example screen shots related to license agreements according to an embodiment of the invention;

25           FIGS. 87A and 88-90 illustrate example screen shots related to royalty statements according to an embodiment of the invention;

FIGS. 95-99 illustrate example screen shots related to payments according to an embodiment of the invention;

FIG. 108 illustrates an example screen shot of a reports view according to an embodiment of the invention;

FIGS. 110A-110C, 111A-111D, 112A-112F, 113, 114A-114B, 115A-115C, 116, and 117 illustrate example reports generated by the licensing module according to an embodiment of the invention;

FIGS. 125A and 125B illustrate a flowchart representing an example operational thread of an embodiment of the invention;

FIGS. 126-150 illustrate block diagrams of subsystems of the licensing module according to embodiments of the invention; and

FIGS. 151-163 illustrate additional use case diagrams representing additional functions supported by the present invention.

In the following text, reference is sometimes made to existing U.S. patents. Also, some of the figures reference or illustrate existing U.S. patents. For illustrative purposes, information from and/or about these patents has sometimes been modified or created in order to support the particular examples being discussed. Accordingly, the information provided herein about these existing U.S. patents should be considered to be fictional unless verified through comparison with copies of the actual U.S. patents that are available from the U.S. Patent and Trademark Office.

## ***Detailed Description of the Preferred Embodiments***

### ***Overview of the Invention***

The present invention is directed to a system, components of the system, a method, components of the method, and a computer program product for patent-centric and group-oriented data processing. Such processing includes, but is not limited to, reporting, analyzing, and planning.

FIG. 1 is a conceptual representation of the invention. The present invention processes patent information 104, which is herein defined to include (but not limited to) U.S. and non-U.S. patents (text and/or images) and post issuance documents (such as Certificates of Correction), and patent-related information, which includes information about patents (herein called patent bibliographic information). Accordingly, the processing performed by the invention is said to be "patent-centric" or "patent-specific."

More generally, the present invention processes any documents, some of which are related to patents, and others which are unrelated to patents. These documents are preferably of interest to a business entity, and include contracts, licenses, leases, notes, commercial papers, other legal and/or financial papers, etc., as well as patents.

For illustrative purposes, the invention is often described herein with respect to patents. However, it should be understood that the invention is also applicable to all types of documents, and the structures, functions, and operations described herein are applicable to all types of documents, whether patent or non-patent.

The present invention also processes other information, preferably business-related information, including (but not limited to) research and development (R&D) information 106, financial information 116, patent licensing information 114, manufacturing information 108, and other relevant business information 110 (which may, for example, include human resources information). This other information is generally called non-patent information (since it includes documents other than patents and may further include information from operational and non-operational corporate databases).

The present invention is adapted to maintain and process massive amounts of documents (several hundred thousand or more). It is often necessary to maintain and process this large number of documents in order to develop strategic, patent-related business plans for the customer.

According to the present invention, processing of the patent information 104 can be conducted either with or without consideration of any of the other information 106, 116, 114, 110, 108.

5 The present invention is group enabled. According to the present invention, a group is a data structure that includes a collection of patents. The patents in a group typically follow a common theme or characteristic (although this is not a mandatory requirement of groups). For example, a first group may include patents that map to a product being manufactured and sold by a company. A second group may include patents that map to a product or product feature  
10 being considered for future manufacture and sale by a company. A third group may include patents owned by a corporate entity. A fourth group may include patents each having a particular person named as an inventor. A fifth group may include patents owned by a competitor. A sixth group may include patents related to a research project. A seventh group may include licensed patents. An  
15 eighth group may include patents and/or non-patent documents related to a litigation in which the customer is involved or has an interest (such a group is also herein called a case). A ninth group may include patents and other documents arbitrarily selected by a customer.

20 The present invention is capable of automatically processing the patents in a group, or the patents in multiple groups (alternatively, the invention can automatically process a single patent). Accordingly, the present invention is said to support "group-oriented" data processing.

As noted above, according to the present invention, processing of the patent information 104 is conducted with consideration of other information 106,  
25 116, 114, 110, 108, called non-patent information. The process of assigning patents to groups is an example of processing patent information with non-patent information. This is the case, because groups are often created according to non-patent considerations. Accordingly, any subsequent processing of the patents in a group involve, by definition, non-patent considerations.

A group may also contain non-patent documents. In fact, a group may contain only non-patent documents. Accordingly, a group is more generally defined as a collection of documents (such as patent documents only, non-patent documents only, or a combination of patent and non-patent documents). The documents in a group typically follow a common theme or characteristic (although this is not a mandatory requirement of groups). Referring to FIG. 1, the invention processes document information 104 alone, or in conjunction with other information 106, 116, 114, 110, 108 (which may or may not be related to the documents). Accordingly, the processing performed by the present invention is more generally described as being document-centric and group-oriented.

### *Components of the Invention*

FIG. 2 is a block diagram of a system 202 according to an embodiment of the invention. The system 202 includes a plurality of databases 216 that store patent information and other information, such as R&D (research and development) information, financial information, licensing information, manufacturing information, HR (human resources) information, and any other information that may be pertinent to the analysis of the patent information. The terms "database" and "table" are used synonymously herein.

An enterprise server 214 accesses and processes the information in the databases 216. In particular, the enterprise server 214 includes modules that are capable of automatically accessing and processing the information in the databases 216 in a patent-centric (or document-centric) and group-oriented manner. These modules are also capable of automatically accessing and processing the information in the databases on a patent by patent basis ("one patent at a time"). Such processing includes, but is not limited to, reporting, analyzing, and planning.



The system 202 preferably includes two types of clients, network clients 206 and web clients 204. These clients 204, 206, pursuant to instructions from human operators or users (not shown), interact with the enterprise server 214 to access and process the information in the databases 216. For example, the clients 204, 206 may request that the enterprise server 214 retrieve certain information, or automatically analyze certain information. The enterprise server 214 performs the requested tasks, and sends the results to the requesting clients 204, 206. The clients 204, 206 present these results to their respective operators, and enable the operators to process the results.

In an embodiment of the present invention, the components of the present invention shown in FIG. 2 are implemented using well known computers, such as a computer 702 shown in FIG. 7. The computer 702 can be any commercially available and well known computer capable of performing the functions described herein, such as computers available from International Business Machines, Apple, Silicon Graphics Inc., Sun, HP, Dell, Compaq, Digital, Cray, etc.

The computer 702 includes one or more processors (also called central processing units, or CPUs), such as a processor 706. The processor 706 is connected to a communication bus 704. The computer 702 also includes a main or primary memory 708, preferably random access memory (RAM). The primary memory 708 has stored therein control logic 710 (computer software), and data 712.

The computer 702 also includes one or more secondary storage devices 714. The secondary storage devices 714 include, for example, a hard disk drive 716 and/or a removable storage device or drive 718. The removable storage drive 718 represents a floppy disk drive, a magnetic tape drive, a compact disk drive, an optical storage device, tape backup, ZIP drive, JAZZ drive, etc.

The removable storage drive 718 interacts with a removable storage unit 720. As will be appreciated, the removable storage unit 720 includes a computer

usable or readable storage medium having stored therein computer software (control logic) and/or data. The removable storage drive 718 reads from and/or writes to the removable storage unit 720 in a well known manner.

5           Removable storage unit 720, also called a program storage device or a computer program product, represents a floppy disk, magnetic tape, compact disk, optical storage disk, ZIP disk, JAZZ disk/tape, or any other computer data storage device. Program storage devices or computer program products also include any device in which computer programs can be stored, such as hard drives.

10           In an embodiment, the present invention is directed to computer program products or program storage devices having software that enables the computer 702 to perform any combination of the functions described herein.

15           Computer programs (also called computer control logic) are stored in main memory 708 and/or the secondary storage devices 714. Such computer programs, when executed, enable the computer 702 to perform the functions of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor 706 to perform the functions of the present invention. Accordingly, such computer programs represent controllers of the computer 702.

20           The modules of the invention discussed herein, such as the grouping module 312, the analysis modules 316, etc., preferably represent software executing in the computer 702.

          The computer 702 also includes a display unit 722, such as a computer monitor, and one or more input devices 724, such as a keyboard, a mouse, other pointing devices (such as a light pen and trackball), etc.

25           The computer 702 further includes a communication or network interface 726. The network interface 726 enables the computer 702 to communicate over communication networks, such as networks 208 and 212, which preferably use the well known HTTP communication protocol.

### *Databases*

FIG. 4 illustrates the databases 216. According to the present invention, the databases 216 store document information (that includes patent information) and information pertinent to the analysis of the document information.

5           FIG. 4 illustrates a particular embodiment of the databases 216, and also illustrates a particular embodiment of the types of tables that the databases 216 contain, and the attributes in the tables. It should be understood, however, that the invention is not limited to the particular database embodiment of FIG. 4. Instead, the invention is adapted and intended to cover other database structures and organizations that are capable of storing document information and  
10           information pertinent to the analysis of the document information. The particular information that is stored in the databases is implementation dependent and varies based on a number of factors, including the type of analysis that is desired, the specific needs of the customer, the type and content of the information that the  
15           customer maintains, etc.

Many of the databases 216, such as the BOM databases 426, the inventor databases 428, and corporate entity databases 430, the financial databases 438, the person databases 432, and the employee databases 434, are initially loaded using information provided by the customer. Such information includes R&D  
20           (research and development) information, financial information, licensing information, manufacturing information, HR (human resources) information, and any other information that may be pertinent to the analysis of the customer's patents and other relevant documents. After initial loading, these databases 216 are updated as necessary to reflect changes in the customer's information.

25           Other information, such as information for the patent bibliographic databases 404 and the patent database 414, may be loaded using information provided by a third party provider, such as a third party provider that specializes in the provision of patent information in electronic form. One such third party

provider is SmartPatents Inc. (SPI) of Mountain View, CA. The patent bibliographic databases 404 may be periodically updated through a subscription service from such third party providers. Similarly, the patent database 414 may be augmented through as-needed orders to the third party providers. It should be understood that the present invention works equally well with data provided by any party as long as the data's format matches the formats of the patent bibliographic databases 404 and the patent database 414.

### *Document Databases*

The document databases 412 preferably include electronic representations of documents of interest to the customer. The document databases 412 represent the customer's repository of documents, and are thus also called the customer's document repository. (The "repository" could alternatively represent all documents represented in the databases 216, whether represented in the document databases 412 or the bibliographic databases 402.)

For example, the patent database 414 includes electronic representations of U.S. and foreign patents of interest to the customer. These patents may be patents owned and/or licensed by the customer, patents owned and/or licensed by competitors of the customer, patents that the customer is considering acquiring, patents that, for whatever reason, the customer is studying, etc. The patent database 414 represents the customer's repository of patents, and is thus also called (in some embodiments) the customer's patent repository.

The patent database 414 preferably has stored therein an image file and a text file for each patent represented in the patent database 414, where the image file and the text file are representations of the patent. Details of an embodiment of the image file and the text file are described in U.S. Patent No. 5,623,681 and U.S. Patent No. 5,623,679.

5 The document databases 412 also include electronic representations of other documents of interest to the customer, such as depositions, pleadings, and prior art references. These documents are respectively stored in a deposition database 418, a pleadings database 416 (generally, pleadings are papers filed with a court), and a prior art database 420. Text and/or image representations of these documents may be stored. These documents may be pertinent to a patent litigation that the customer is involved in.

10 The documents in the document databases 412 may be text, images, graphics, audio, video, multimedia, and/or any other information representation that can be stored in electronic form.

15 It should be understood that the document databases 412 of FIG. 4 are shown for purposes of illustration, and not limitation. As mentioned above, the document databases 412 store electronic representations of documents that are of interest to the customer. Accordingly, the types of document databases 412 and the contents of the document databases 412 are, by definition, customer and implementation specific.

### *Document Bibliographic Databases*

20 The document bibliographic databases 402 store information about documents (as opposed to the documents themselves). More particularly, the document bibliographic databases 402 store bibliographic information about documents.

### *Patent Bibliographic Databases*

25 The patent bibliographic databases 404 store bibliographic data about U.S. and non-U.S. patents. Such patent bibliographic data includes, but is not limited to, the information on the front page of patents, such as: the patent number, the

issue date, the inventors, the title, the assignee, the serial number, the filing date, the U.S. and international classifications, the fields of search, the references cited, the primary examiner, the assistant examiner, the attorney, the agent, the law firm, priority information, related application information, the number of claims, the number of drawing pages, the patent term, the expiration date, etc. The patent bibliographic databases 404 can also include one or more user defined fields that can store large amounts of data, such as 32 Kbytes or more of data.

In an embodiment of the invention, the patent bibliographic databases 404 store bibliographic information on all U.S. patents. In other embodiments of the invention, the patent bibliographic databases 404 store patent bibliographic information on a subset of all U.S. patents, such as all U.S. patents that are available in electronic form from the U.S. Patent Office, or all U.S. patents that issued after a certain date.

#### *Other Document Bibliographic Databases*

The document bibliographic databases 402 include store bibliographic information on other types of documents that are of interest to the customer. For example, if the customer is interested in depositions, pleadings, or prior art references, then the document bibliographic databases 402 would store bibliographic information on depositions, pleadings, or prior art references in deposition bibliographic databases 406, pleadings bibliographic databases 408, and prior art bibliographic databases 410, respectively.

#### *Notes Database*

The present invention supports annotation of the documents in the document databases 412. More particularly, the present invention allows users to create and link annotations (also called notes) to any portions of the documents

in the document databases 412. Such annotations can include text, graphics, images, video, audio, and/or any other information representation that can be stored in electronic form.

5       The present invention also allows various information to be stored with annotations, such as the date of creation, the creator, the date of modification, a note title and/or subject, access rights, etc.

Embodiments of the notes databases 440 are described in U.S. Patent No. 5,623,679 and U.S. Patent No. 5,623,681.

### *Groups Databases*

10       Information on groups is stored in the group databases 421. Generally, a group is a data structure that includes any number of documents that typically follow a common theme or characteristic (although this is not a mandatory requirement of groups). More particularly, a group is a data structure that includes any number of patents that typically follow a common theme or  
15       characteristic (although, again, this is not a mandatory requirement of groups). Groups are document-centric, or in many cases, patent-centric.

There are two classes of groups: predefined groups (also called system defined groups) and user-defined groups (also called arbitrary groups).

20       Patents (and/or documents) in predefined groups follow a predefined theme or characteristic. Database tables, fields, and attributes of a predefined group are specific to the predefined theme/characteristic of the predefined group. Accordingly, different predefined groups have different database tables, different database fields, and different database attributes. Information on predefined groups is stored in the predefined or system defined group databases 422.

25       Patents (and/or documents) in user-defined groups may or may not follow a common theme or characteristic. Any theme or characteristic that they do

follow is defined by the user. Accordingly, user-defined groups are also called arbitrary groups.

5 All user-defined groups have the same, generic database tables, fields, and attributes. However, users may elect to use these database tables, fields and attributes differently for different user-defined groups. Information on user-defined groups is stored in the user-defined group databases 424.

### ***Enterprise Server***

10 The enterprise server 214 is preferably implemented as one or more computers (such as the computer 702 shown in FIG. 7) each having at least 128 MBytes of main memory 708 and running Microsoft Windows NT. The enterprise server 214 could, alternatively, be implemented using other memory configurations, and other operating systems, such as (but not limited to) UNIX, Windows 95, MS-DOS, the Apple Operating System, etc. Accordingly, the specific hardware and software implementations discussed herein are provided  
15 for purposes of illustration, not limitation (this applies to all specific hardware and software implementations discussed herein, both for the enterprise server 214 and for other components of the invention). The invention can utilize any hardware, software, and operating system capable of performing the functions described herein.

20

### ***Document Storage and Retrieval Module***

The document storage and retrieval module 308 in the enterprise server 214 stores and retrieves documents from the document databases 412.

### ***Notes Module***



The notes module 314 manages and interacts with the notes databases 440.

The client notes module 514 enables a user to view and manipulate notes.

### *Searching Module*

5           The searching module 310 in the enterprise server 214 interacts with a search engine 324 to conduct searches through the data in the databases 216 pursuant to search requests from the clients 204, 206.

### *Grouping Module*

10           The grouping module 312 in the enterprise server 214 manages and interacts with the group databases 421.

### *Analysis Modules*

The analysis modules 316 are shown in FIG. 6. These analysis modules 316, which are also called methodology modules 316, automatically interact and process data contained in the databases 216 pursuant to user commands.

### 15           *Databases*

20           The database structures of the document bibliographic databases 402, the group databases 421, the person databases 432, the employee databases 434, the security databases 436, the financial databases 438, and the methodology support databases 442 are shown in FIGS. 8B-8M. These figures also depict the interaction and connection between these databases. FIG. 8A illustrates the preferred orientation of FIGS. 8B-8M with respect to one another.

It should be understood that the tables and attributes shown in FIGS. 8B-8M only represent one embodiment of the present invention. The data in the databases 216 could be stored using other combinations of tables and attributes. Such other combinations of tables and attributes will be apparent to persons skilled in the relevant arts based on the discussion contained herein. Accordingly, the tables and attributes are shown in FIGS. 8B-8M only for purposes of illustration, and not limitation.

### ***Financial Module***

The financial modules 610 in the enterprise server 214 perform patent-centric and group-oriented processing of the data in the financial databases 438. Examples of the functions performed by the financial modules 610 include determining the research and design (R&D) expenditures on a product or product line basis, determining the R&D expenditures per inventor or per employee on a product or product line basis, determining net licensing revenue on a product or product line basis, determining the number of patents issued on a product or product line basis, determining patent maintenance fees on a product or product line basis, determining market share on a product or product line basis, determining the tax rate on a product or product line basis, determining marketing costs on a product or product line basis, determining selling costs on a product or product line basis, determining the number of outstanding shares (P/E) on a product or product line basis, determining revenue on a product or product line basis, determining cumulative product revenue on a product or product line basis, etc. The financial modules 610 can also perform the above processing on a geographical region basis, or on a time basis.

***Console***

FIG. 9 illustrates a console according to an embodiment of the invention. The console 902 includes a first window or pane 904, a second window or pane 906, and a third window or pane 908. The first pane 904 is also called the group pane 904, the second pane 906 is also called the document pane 906, and the third pane 908 is also called the notes pane 908.

A group hierarchy is depicted in the group pane 904. In the example of FIG. 9, the top level or root group in the group hierarchy is called the repository group 910. The child groups of the repository group 910 are not shown in FIG. 9 (i.e., the operator has not expanded the repository group 910 in the group pane 904). The child groups of the repository group 910 are shown in FIG. 133 (i.e., the operator has expanded the repository group 910 in the group pane 904 in the example of FIG. 10).

Referring again to FIG. 9, the document pane 906 includes a list of patents and other documents which are contained within a group selected from the group hierarchy depicted in the group pane 904. The patents and documents are listed in a tabular or "spreadsheet" format. The list of patents and documents in the document pane 906 includes both the patent numbers and patent bibliographical information for the patents, and bibliographic information for the non-patent documents. Such patent bibliographic information displayed in the document pane 906 includes the title, abstract, inventor, class, issue date, and user-defined keywords. All additional patent bibliographic information can be viewed in the document pane 906 by utilizing the horizontal scroll bar 920 to sideways scroll in the document pane 906. Other embodiments of the invention allow the user to select an arbitrary number of bibliographic fields to view. In example of FIG. 9, no patents are listed in the document pane 906 because a group has not been selected in the group hierarchy depicted in the group pane 904.

The notes pane 908 displays a list of the notes associated with either a group selected in the group pane 904, or a patent or document selected in the document pane 906. The list of notes in the notes pane 908 is presented in a

5 tabular or "spreadsheet" format. The list of notes in the notes pane 908 includes information that identifies the type of the note (that is, either a patent/document note or a group note), and the title of the note. All other bibliographic or other information relating to notes can be viewed by manipulating the horizontal scroll bar 922 in order to sideways scroll in the notes pane 908.

### *Licensing Module*

10 As shown in FIG. 11A, an embodiment of the invention includes a licensing module 1102 (also herein called the licensing system 1102). The licensing module 1102 is also referred to herein as the SmartPatents Prism™ system.

15 Customers use the licensing module 1102 to manage their intellectual property (IP) assets for maximal value through the creative use of licensing. The licensing module 1102 provides the tools a licensor needs to manage its licensing effectively through tracking a variety of objects, including but not limited to out-licenses (i.e., licenses where the corporate entity is the licensor), in-licenses (i.e., licenses where the corporate entity is the licensee), licensing parties (i.e., any parties involved with a license agreement, such as the licensee(s), the licensor(s), license agents, license organizations, attorneys, etc.), royalty statements, royalty payments, etc.

20 Customers also use the licensing system 1102 to better understand how they are making strategic use of their IP assets and to audit the licensees' contribution to the value of those assets.

25 As shown in FIG. 12, in an embodiment of the invention the licensing module 1102 is a standalone system, existing and operating independently of the enterprise server 214 and the enterprise databases 216. In this standalone configuration 1201, the licensing module 1102 does not access data in enterprise/IPAM database 216. Instead, the licensing module 1102 utilizes data

in its own databases, i.e., a licensing database 1204 and a core database 1208, among potentially others. These databases are described in detail below. (The invention is not limited to the arrangement of data described herein. In other embodiments, the data and attributes described herein are stored in combinations of databases and tables other than that described herein.)

As shown in FIG. 13, in another embodiment, the licensing module 1102 interacts with the enterprise server 214 and/or the enterprise databases 216, such that users of the licensing module 1102 may access and utilize groups and/or IP assets, as well as other information, stored in the enterprise databases 216, and/or may interact with the enterprise server 214 to further manage groups of IP assets and/or other objects that are being tracked through the licensing module 1102.

According to some embodiments of the invention, the enterprise server 214 is herein also sometimes referred to as the Intellectual Property Asset Manager™ (IPAM™) server 214.

The database architecture associated with the licensing module 1102 includes a number of databases. As noted above, the standalone embodiment of the licensing module 1102 (shown in FIG. 12) includes a Licensing database 1204 and a Core database 1208.

The IPAM-integrated embodiment of the licensing module 1102 (shown in FIG. 13) includes the Licensing database 1204. The Core database 1208 is substantially or completely replaced with the enterprise database 216 (also called the IPAM database 216). The Core database 1208 contains standalone versions of all of or at least a subset of the tables included in the IPAM Server database 216. Accordingly, when the IPAM database 216 is available, there is little or no need for the core database 1208.

In an embodiment, the licensing module 1102 is implemented as a two-tier client/server model. In an alternate embodiment, the licensing module 1102 is implemented as a three-tier model using standard middleware technology such as CORBA or COM along with technologies compatible with them, including the

appropriate security manager for the application server middleware layer. The invention is not limited to these embodiments. The architecture preferably provides a thin-client C++ component, a secure application domain server, and a secure database server (such as a SQL Server), linking the latter two components with ODBC for maximal portability. The user interface and object model are tightly integrated and use well known component technologies such as ActiveX and DAO. In an embodiment, security relies on defining SQL Server database users with passwords and appropriate privileges on the database objects.

An example thread of operation of the licensing module 1102 shall now be generally described. At initial set up of the licensing module 1102, and periodically thereafter (or as the circumstances dictate), a License Administrator and a System Administrator must perform various set up tasks, such as entering territories, fields of use, and currency conversion intervals.

After initial set up, the customer enters IP business related information, such as but not limited to existing license agreements, new license agreements, and/or related objects, such as but not limited to assets, asset packages, contacts, compensation terms, and so on. Specifically, in an embodiment, the Data Entry Clerk enters assets, and the License Administrator creates asset packages to package assets for licensing. The Data Entry Clerk enters contacts by entering organizations, people, and contact methods. Then the Data Entry Clerk enters the license information. At that point, the License Administrator takes over to modify the license agreement with more complex data that requires an understanding of licensing.

As data grows, the License Administrator begins querying the system interactively in response to questions and issues that arise. He or she will also start generating reports. At some point, an auditor or other interested party will also query and generate reports. The License Administrator then may need to do some maintenance on the agreements, accompanied by occasional maintenance

by the IPAM Administrator on objects that require Administrator privileges to remove.

As time goes on, licensees submit royalty statements and payments. The Data Entry Clerk enters these. The License Administrator then allocates the payments to expected revenue and to royalty statement details. This linking  
5 enables the Licensing module 1102 to generate meaningful revenue reports.

More generally, the licensing module 1102 enables users to manage, track, interact with, process, analyze, etc., intellectual property (IP) related transactions. In a preferred embodiment, such IP related transactions include the licensing of  
10 IP assets, and the management, tracking, interaction with, processing, analysis, etc., of such licensing activities using license agreements, asset packages, royalty statements, payments, etc. However, the invention is not limited to this embodiment. Instead, the invention is intended and adapted to cover the management, tracking, interaction with, processing, analysis, etc., of IP related  
15 transactions using any object, vehicles, data, etc., in accordance with the scope and spirit of the functions described herein.

The Licensing module 1102 includes a number of features, including but not limited to the following:

- Import/Export: Support for loading various kinds of objects into  
20 the Licensing databases from flat files in a standard format to reduce the need for extensive data entry (or, worse, reentry).

- Strategic analysis: Features that help license administrators and marketers to understand the strategic implications of licensing their intellectual property. This includes offering a database of SEC 10K filings of "material"  
25 license agreements on which you can search to determine the range of royalty terms and the strategic issues that arise in a specific industry.

- Market opportunity tracking: Tracking of sales opportunities related to contacts, including inquiries, targeted potential customers, and infringement tracking.

° Marketing Communications and Sales support: Support for customer inquiries, IP marketing brochure querying, and e-commerce sales support.

5       ° Asset types: Support for more asset types, particularly in the arena of Know How. In an embodiment, Know How is a class with various kinds. In alternate embodiments, subclasses are created for the broad areas within Know How such as Technical Data, Technical Assistance, Process, Software, and other such categories with specific attributes.

10       ° Electronic Data Interfacing: support for the ability to transfer royalty statements and payments from licensee to licensor automatically through the Internet or other communications media to improve compliance, simplify auditing, and reduce data entry requirements.

15       ° Licensing In: Extensive support for the licensee, including royalty statement generation, payment due flagging, and sublicense tracking.

20       ° Compliance workflow: Support for the contract administrator in monitoring the compliance of agreement parties to the terms and conditions of the license agreement, such as generating royalty invoices, checking revenue numbers, and detailed representation of contract clauses and their compliance requirements

25       ° Sublicense tracking: The ability to track sublicensing of a license agreement for better royalty tracking and control and contract compliance.

° Contract development workflow: Support for user-defined templates for license agreements and generation of standard agreements with designated clauses.

° International licensing support: support for various issues that arise with international licenses, including support for offset licensing ("counter trade" or "industrial participation" licensing) and the various registration requirements for copyrights, trademarks, and patents in foreign countries and the



import/export requirements of foreign governments that impact royalty and fee payments.

- Custom report integration: the ability to integrate custom reports into the Reports View and run them along with standard reports. This includes integration of a complete report writer into the Licensing module 1102.

- Administrative audit trails: support for object version auditing through recording the user and timestamp for each change to the persistent data.

- Help Desk support: Extensive support for internal customers of the system getting answers to basic questions through query facilities in the Licensing module 1102, ranging from a simple frequently-asked-question style to the ability to query information about licenses, royalty statements, and payments.

- Valuation: Support for alternative methods of valuing a license and its licensed assets.

The licensing module 1102 is described in greater detail in the following sections.

### *User Roles*

Generally, the licensing module 1102 involves a number of user roles, including but not limited to the following. The invention is not limited to these functions being performed by the people specified below. In other embodiments, other people in other user roles can perform the following functions.

- The Data Entry Clerk: a user who enters basic data about contacts, licenses, royalty statements, and license payments. Generally, a Data Entry Clerk does not require much knowledge about licensing or intellectual property.

- The License Administrator: a user who enters more complex information about licenses, royalty statements, and payments and who works with executives, corporate counsels, licensees, and others to provide information about the licenses and revenue in the IPAM system 1102. Generally, a License

Administrator requires extensive knowledge about licensing and licensed intellectual property.

° The Auditor: a user who generates summary and detail reports about the licenses and their associated revenue and who queries information interactively to investigate specific issues. Generally, an Auditor requires extensive knowledge about licensing and the details of accounting for license revenue, but does not require write access to the system.

° The System Administrator: a user who handles security management, data entry of supporting database information, and cleanup of the database on demand from others.

° The Database Administrator: a user who sets up and controls the physical layout of the database that the licensing module 1102 uses.

Generally, the System and Database Administrators have full access to everything in the system. The other users have access granted to them by owners of asset packages or those with write access to security classifications. The different roles above do not imply any particular security access; each user has whatever access they are granted by the Administrator or other users.

FIG. 15 illustrates an object-oriented view of a preferred actor hierarchy 1502 for the licensing system 1102. As FIG. 15 illustrates, all actors inherit the base capabilities of a generic user 1504, who may be a user of the licensing system 1102. This user, for example, could have the authority to view data, conduct searches, and run reports. Other actors, such as the system administrator 1506, the data entry clerk 1508, the auditor 1512, and the database administrator 1516, have capabilities above and beyond that of the user 1504. These additional capabilities are elaborated above, and further indicated below. A super user, the licensing administrator 1518, preferably has complete access to the licensing system 1102, and thus is shown in FIG. 15 as inheriting all the capabilities of all the other actors.

### ***Architecture***

FIG. 11B is a block diagram of the licensing system 1102 according to an embodiment of the invention.

5 There are three general functions of the Licensing module 1102: data entry, data exploration, and reports. Underneath the user interface, there is an object model that implements all the different kinds of licensing objects that Licensing module 1102 needs to support data entry, to provide exploration of the database, and to permit generation of effective reports. Licensing module 1102 preferably connects to a SQL Server database server 1140 through ODBC 1138  
10 to provide persistent storage of its data and to support third-party report definitions.

### ***User Interface***

The user interface preferably includes the following components.

#### ***Licensing System User Interface Module***

15 The licensing system user interface module 1101 is the primary user interface for the licensing related functions of the licensing system 1102 described herein. The licensing system user interface module 1101 interacts with users via the windows and dialogs described herein. Preferably, the licensing system user interface module 1101 is written in Visual C++ and MFC (Microsoft  
20 Foundation classes, a well known class library for Visual C++), although the invention is not limited to this implementation.

#### ***Licensing System Administrator User Interface Module***

The licensing system administrator user interface module 1104 is the user interface to the licensing related administrative functions described herein. The licensing system administrator user interface 1104 is preferably written in Visual C++ and MFC, although the invention is not limited to this implementation.

5

### *Data Entry*

In an embodiment, operation of the licensing module 1102 is based on license agreements. (The invention is not limited to this embodiment. Specifically, in other embodiments, the licensing module 1102 is based on other IP related transactions, such as assignment agreements, technology transfer agreements, security interests, or any other agreement involving a transaction that includes or relates to an IP asset.)

Generally, license agreements are completely nonstandard in text format, but customers need to track specific, standard information about the agreement: its terms, its duration, its parties, the payments and statements customers receive, etc. They need to explore how their IP assets are generating revenue. They need to be able to license IP assets including patents, trademarks, copyrights, know how, trade secrets, other licenses, etc. Customers need to package multiple assets for licensing through a single license. They need to report on different aspects of their license portfolio. They need to search the document text to identify agreements that they can use as templates for further agreements.

20

Consequently, the Licensing module 1102 enables users to enter structured data about license agreements, royalty statements, payments (this information is preferably stored in the licensing database 1204). The licensing module 1102 also enables users to enter data about companies and people to which IP assets can be licensed (this information is preferably stored in the licensing database 1204). The licensing module 1102 also enables users to enter IP assets (patents and other types) (this information is preferably stored in the

25

licensing database 1204). This part of the licensing system 1102 can link to the IPAM Server 214 for access to the IP assets and groups it supports. The licensing module 1102 also adds to those IP assets as necessary to round out the range of asset types that customers wish to license: trademarks, copyrights, and trade secrets as well as patents. Generally, information on non-patent assets are stored in the licensing database 1204 or the core database 1208. Information on patent assets are stored in the IPAM database 216 when working in the IPAM integrated mode (FIG. 13), and stored in the licensing database 1204 or the core database 1208 when working in the standalone mode (FIG. 12). The Licensing module 1102 also provides asset packaging options beyond lists of assets, including search packages based on IPAM search groups and descriptive asset packages that define asset groups with text rather than enumerating assets.

### *Licensing Reports*

Licensing reports are an important feature of the Licensing module 1102. Customers use reports generated by the licensing module 1102 to track the revenue due from licenses, to compare that expected revenue to the actual revenue received, and to evaluate the success of license management in their business units. The invention also supports other reports.

Preferably, the report generation module 1106 is responsible for generating the reports requested by users. The report generation module 1106 generates reports using the data in the licensing related databases, including printing the contents of each major object in the system (Print License, Print Asset, and so on). Preferably, the report generation module 1106 connects to the databases through ODBC 1138 or through the Microsoft SQL Server driver, for example. The report generation module 1106 can be implemented using a commercial report generation product, such as Crystal Reports, although the invention is not limited to that embodiment.

In embodiments where the report generation module 1106 is implemented using a commercial report generation product, a report generator interface 1132 interacts with the report generator 1106 to cause the report generator 1106 to generate reports per user instructions. For example, the generator interface 1132 may interact with the report generator 1106 in a manner consistent with the API (application program interface) of the report generator 1106 to cause the report generator 1106 to access the databases 1608 for appropriate information, and to process and format that information in reports, per user commands.

### *Data Exploration Module*

Strategically, licensing managers will use the Licensing module 1102 to access license data in creative ways.

A strategic use of the licensing system 1102 involves ad-hoc query functions in combination with reports to identify IP assets that are underperforming or outperforming and to cross-reference between license agreements, licensees, royalty terms, royalty statements, and royalty payments. The ad-hoc query facility uses the same user interface that provides data entry facilities in an easy-to-use query-by-example format.

The ad-hoc query facilities also support licensing personnel and auditors in interactive tactical tracking down of issues such as payments due, payments not received, royalty statement details, and errors in data entry.

### *Object Model*

The object model of the Licensing module 1102 represents the transient aspect of the persistent data in the database server. The subsystems of the object model provide interfaces to the various licensing-related objects and higher-level objects that aggregate them for use in the Licensing module 1102. The object

model supports a number of subsystems and/or object/data types, including but not limited to the following:

- ° Entity 1112: The people and corporate entities that make up the licensors, licensees, and other parties to the license agreement.

5

- ° Contact: The contact-related information for entities.

- ° Role: The relationships between entities and other subsystems.

- ° Agreement 1114: The licensing agreement, including basic royalty terms and advanced royalty structures.

10

- ° Package 1118: IP assets and their groupings, including enumerated groups of license agreements, groups based on a search condition from the IPAM Server, and descriptive groupings.

- ° Royalty 1120: The royalty statement that details the responsibility of the licensees for royalty payments.

15

- ° Payment 1122: The payments received from entities related to licenses and their links to the general ledger.

More particularly, the licensing system domain 1108 is an object layer called a licensing system domain 1108 that represents a series of subsystems (generally indicated as 1148 in FIG. 11B) that contain objects that the licensing system 1102 requires to provide the functionality described herein. Preferably, each subsystem design minimizes the connections to other subsystems from its internal objects with the objective of maximal reusability as components. Each subsystem provides a complete interface to the subsystem and its component objects. Each subsystem is preferably a separate DLL (dynamic link library) and is preferably contained within a separate C++ namespace.

20

25

Each subsystem exports a set of Transaction subclasses, where a transaction is a Database subsystem component that provides support for database transactions. Each transaction subclass represents a transaction use case. Some transaction classes provide methods for generating objects either as new objects or as objects queried from the database. Others provide the ability to update and

delete objects from the database. Each use case is a single transaction through the system, and each must execute entirely within a single thread to ensure thread safety. The use cases supported by an embodiment of the invention are described in sections below.

5           The user interface accesses the transaction subclasses as the primary interfaces. Transactions correspond to the work the user does with those objects (entering new assets, adding payments to a license, and so on). Block diagrams of the subsystems are illustrated in FIGS. 126-150. Details on these subsystems will become apparent from the following discussion of use cases.

10           It is noted that allocation of functions to the subsystems 1148 may vary according to the implementation. Accordingly, the subsystems shown in FIG. 11B should be considered only an example.

#### ***Data Access Module***

15           The data access module 1110 provides an interface between system databases 1608 and the modules of the licensing system 1102. Preferably, the data access module 1110 includes the Microsoft Active Data Objects COM component (preferably version 1.5) that provides a SQL interface for accessing various data sources. In this case, it connects to ODBC 1138.

#### ***Database Architecture***

20           The database architecture preferably includes a Database Server 1140 (such as a SQL Server or an Oracle server) that preferably manages the persistent data as a set of relational tables. The databases 1608 in the database architecture each preferably includes a number of databases, each containing a well-defined and reusable set of data components. (It is noted that the invention is not limited



to this arrangement of data. In other embodiments, the data described herein is stored in database arrangements other than that described below.)

5           °       The Licensing database 1204 contains the Role, Agreement, Royalty, and Payment data. More particularly, the licensing database 1204 contains tables that provide information about licensing-related objects (agreements, royalties, payments, currency, and so on). The Licensing database 1204 also contains the standalone asset data. The Licensing Database 1204 further includes tables supporting time periods (start date/end date type relationships). The licensing database 1204 also contains the Entity and Contact data, i.e., data supporting people, organizations, and contact information for them. 10       The Licensing database 1204 provides a series of views for tables in all the other databases. The domain and the report definitions access the Licensing database 1204, using the views to access data in the other databases. This provides a single, unified database interface for the higher level layers in the licensing system 1102. Views in the Licensing database 1204 are preferably integrated 15       when necessary to provide easier access to complexly related tables. The objective in dividing the tables between several databases is to separate the data for reuse in other applications.

20           °       The Core database 1208, which contains Group data (that is, information on groups and the assets in each of the groups) and patent asset data (either downloaded from the IPAM Server database 216 or a standalone version of that database 216 that Licensing module 1102 uses). More generally, the Core database 1208 contains exactly the same or, alternatively, substantially the same tables that are in the IPAM database 216 (optionally, any databases in the IPAM 25       database 216 that are not accessed by the licensing system 1102 are not included in the core database 1208).

      The standalone version of Licensing module 1102 has a stand-in version of the IPAM database 216 (i.e., the Core database 1208) that, in an embodiment, contains only some of the tables in that database that the module uses (for

example, group, document, and a few miscellaneous tables). The patent tables in the IPAM Server database 216 also replaces the patent tables (but not non-patent asset tables) in the licensing database 1204 or the core database 1208 for the integrated version of the Licensing module 1102 (FIG. 12).

5           FIG. 16 illustrates a conceptual, object oriented view of the databases 1608. The licensing database 1204, the core database 1208, and the IPAM database 216 inherit all the abilities of a generic database 1604.

***Licensing Module As a Standalone Module or Integrated with IPAM  
(IPAM Integrated)***

10           As discussed above, in one embodiment the Licensing module 1102 integrates with the IPAM Server 214 through the IPAM database 216, which it accesses directly.

          The standalone version of the Licensing module 1102 replaces the IPAM database 216 with a Core database 1208 that emulates the IPAM tables 216 that the Licensing module 1102 uses, including the Group-related tables, the Document table, the IP Document table hierarchy, and the UDD (unique document descriptor) tables. See FIGS. 8A-8M.

15

          With IPAM integration, the Licensing module 1102 has all the patent and UDD (user defined documents) documents available as assets and all groups available for group asset packages from the IPAM database 216. With the standalone version of the Licensing module 1102, the customer must enter patents through data entry screens in the Licensing module 1102 interface (in an embodiment, these are not present when the system is integrated).

20

          Preferably, the Licensing module 1102 integrates with the IPAM security system through the IPAM security broker, a library that is part of the IPAM server technology.

25

*Object Display Window of the User Interface of the Licensing module*

FIG. 14 illustrates an object display window 1402 of the user interface of the Licensing module 1102. The object display window 1402 includes a number of elements:

- 5           °       Menus: At the top of the object display window 1402 are a variety of pull-down menus 1406. These menus 1406 allow easy access to all of the transactions you can accomplish with the Licensing module 1102
- °       Tool Bar: Under the pull-down menus 1406 is a tool bar 1404, which presents a line of often-used tools, such as New, Save, Print, and Find.
- 10          °       Licensing Functions Tool Bar: Along the side of the object display window 1402 is a licensing functions tool bar 1408 that include icons providing access to the major objects in the Licensing module 1102. These icons include a license agreements icon 1412, a contacts icon 1414, an asset packages icon 1416, a royalty statements icon 1418, a payments icon 1420, a reports icon 1422,
- 15          and a recycle bin icon 1424. Clicking on one of these icons changes the content of the content view area 1426 so as to display the indicated kind of objects and the various components that make them up.
- °       Content View area: The content view area 1426 is the main area of the object display window 1402. The content view area 1426 contains one or
- 20          more controls for managing the objects that make up Licensing module 1102. These panes include spreadsheet views, and one or more "explorer" or hierarchical views for organizational structure. A spreadsheet view lets you configure the columns to display and sorts on a column when you click on the header.
- 25          °       Status Bar: A status bar 1410 appears at the bottom of the object display window 1402. The status bar 1410 displays application status, short help messages, keyboard status (CAPS LOCK, NUM, etc.), etc.

5           °       Recycle Bin: If you remove one of the major objects (entity, asset, asset package, license, royalty statement, payment, currency, etc.), the object gets recorded in the Recycle Bin 1424. You can either restore individual objects in the Recycle Bin view, or you can empty the recycle bin 1424 to make the changes permanent. An object in the recycle bin 1424 is preferably not visible anywhere else in the application.

10           °       Dialogs: There are various dialogs that pop up (not shown in the example of FIG. 14), usually as a result of clicking on a tool button or double-clicking on an object in a spreadsheet; the dialogs accomplish data entry, data modification, searching, etc.

### *Operational Features of the Licensing Module*

Operational functions supported by embodiments of the licensing module 1102 are described in detail in the following sections.

15           Preferably, these functions are implemented by one or more computers operating according to control logic, or software. Such software is preferably written in an object oriented manner. Accordingly, the licensing module 1102 can be said to be an object oriented system.

20           Use cases are sometimes used to describe the manner in which an object oriented system works. Generally, a use case describes a transaction processed in an object oriented system.

          Use cases often include extensions. Extensions are conditional. For example, the enter contact method use case described below extends the enter entity method use case. In other words, under certain conditions when you are entering an entity, you will enter a contact method.

25           In the following section, the functions supported by embodiments of the licensing module 1102 are described by describing use cases that correspond to these functions. Coding of software to achieve these functions will be readily

apparent to persons skilled in the relevant arts, based on the description contained herein, including but not limited to the following description of use cases. In the following, use cases are described as being implemented as procedures, but in practice the functionality of any number of use cases can be achieved through one or more software procedures.

An embodiment of the licensing system 1102 is shown in FIG. 17A. Each of the ovals shown in FIG. 17A corresponds to a logical functional module of the licensing system 1102. Generally, these functional modules correspond to the use cases that are described below. The relationships between the functional modules are indicated in FIG. 17A.

The invention also preferably includes a licensing administration module 1701 (FIG. 17B). The licensing administration module 1701 performs administrative tasks associated with the licensing system 1102. The licensing administration module 1701 is preferably accessible only by administrators.

A functional module may "use" another functional module. In other words, in the course of performing its specified function, a functional module may call or invoke the services of another functional module. Alternatively, a functional module may "extend" another functional module. In other words, a functional module may extend the capabilities of another functional modules. In some cases, not all relationships between modules are shown in the figures. The relationships between functional modules are further described in the following sections.

In the following, certain steps are described as being preferably performed by some actors, but in practice and/or in other embodiments, the steps are performed by other actors, including but not limited to those described herein.

### ***General Purpose Use Cases***

There are several general-purpose use cases that apply to all users or to generic objects. These are described in the following sections.

#### ***Login***

5           The login use case 1802 is diagramed in FIG. 18.

          The User enters his or her user name and password and any connection string required by the ODBC data source in a login dialog 1902 shown in FIG. 19. The licensing module 1102 logs the user into the server preferably through the ODBC connection mechanism and authenticates the user through the IPAM security system 1602.

10

#### ***Display Help***

          The display help use case 2002 is diagramed in FIG. 20.

          The User requests help in a specific context or requests display of the help contents, index, or query preferably using a standard help system, such as the Microsoft Windows help system. The display help module 1704 displays the appropriate help screen by accessing, retrieving, and displaying pertinent information from a help file 2004.

15

#### ***Print Object***

          The print object use case is diagramed in FIG. 21.

20           The Auditor or License Administrator selects an object and prints the object in a standard print format to a system printer using the print object module 1706.

The operation of the print object module 1706 is further detailed below:

Step 1: The Auditor or License Administrator (the Actor) starts the read-only transaction by selecting an object (license agreement, asset package, entity, royalty statement, or payment), and then selects the Print command.

5 Step 2: The print object module 1706 displays a Print <Object> dialog, where  
 <Object> is one of the five basic objects from step 1.

Step 3: The Actor sets the printing options, including any printer setup commands, then issues the instruction to print, ending the transaction.

**Step 4:** The print object module 1706 prints the object.

10            If the Actor in step 1 selects a License Agreement, the Print License use case  
extends the print object module 1706.

If the Actor selects an Asset Package, the Print Asset Package module 1710 use case extends the print object module 1706.

15                    If the Actor selects an Entity, the Print Entity module 1708 extends the print  
                      object module 1706.

If the Actor selects a Royalty Statement, the Print Statement module 1714 use case extends the print object module 1706.

If the Actor selects a Payment, the Print Payment module 1716 extends the print object module 1706.

20 *Contacts*

A Contact is an organization or person (general term "entity", which also encompasses users and user groups in the security system). People play roles in organizations, and both people and organizations have contact methods (telephone, mail address, and email).

25 A Contact View 2202 is displayed by selecting the contacts icon 1414 (FIG. 22). The contact view 2202 contains two panes, an explorer view (organization pane) 2204 of organizations and a list of people (people pane) 2206. It is possible to expand

and contract the organization hierarchy in a well known manner. When you select an organization, the people in that organization appear in a people list pane 2206 in a tabular spreadsheet format..

5 Double-clicking on an organization in the explorer view 2204 displays the data modification dialog for organizations. Double-clicking on a person in the people list pane 2206 displays the data modification dialog for people. You can double-click on an empty record or click the New button to create a new organization or person.

The Licensing System Administrator creates, modifies, and/or removes roles from the Licensing Database 1204.

10

### *Enter Entity*

The enter entity use case 2302 is diagramed in FIG. 23.

The Data Entry Clerk enters entity information for people and organizations, including organization levels, people's names, and (optionally) contact methods. The Data Entry Clerk may optionally link a person to an organization with a specific role.

15

The enter entity use case 2302 is further described below:

Step 1: The Data Entry Clerk begins the transaction.

Step 2: The enter entity module 1718 displays the organizations in the organization pane 2204 and people in the people pane 2206 stored in the licensing database 1204 using the Display Entities module 1722.

20

Step 3: The Data Entry Clerk takes any of these actions (for example):

Step 3.1: The Data Entry Clerk adds an organization to the list of organizations.

Step 3.2: The Data Entry Clerk adds a new person to a list of people for an organization.

25

Step 3.3: The Data Entry Clerk copies an existing person into a new organization.



Step 4: The enter entity module 1718 displays a data entry form and creates a new entity unique identifier.

Step 5: The Data Entry Clerk enters an optional description for the new organization/person.

5 Step 6: The Data Entry Clerk commits the transaction.

Step 7: The licensing system 1102 stores the entered entity information in the licensing database 1204 and confirms the commit to the Data Entry Clerk.

There are a number of extensions to the enter entity module 1718.

10       ° For a person, the Data Entry Clerk enters the First Name, Middle Name, and/or Last Name.

° For an organization, the Data Entry Clerk enters the Name. Optionally for an organization, the Data Entry Clerk enters the date on which the organization came into existence.

15       ° Optionally for an organization, the Data Entry Clerk chooses an organizational level from a list.

° If the Data Entry Clerk wants to enter contact information for the entity, the Enter Contact Method use case extends this use case.

20       ° If the Data Entry Clerk is entering a person and wants to link that person to an organizational role, the Link Person to Organizational Role use case extends this use case.

### *Enter Contact Method*

The enter contact method use case 2402 is diagramed in FIG. 24.

25 This use case extends or is used by another use case, which supplies the unique identifier of an entity. The Data Entry Clerk enters one or more contact methods for that entity.

The enter contact method module 1720 is further described below:

Step 1. The using use case passes in an object identifier for an entity with which to associate a contact method.

Step 2. The Data Entry Clerk chooses one of three types of contact method: telephone, email, or mail address.

5 Step 3. The enter contact method module 1720 displays an entry form appropriate for the type of contact method selected and generates a unique priority, defaulting it to one more than the last priority entered for methods associated with this entity:

```
10      SELECT MAX(Priority)+1
      FROM Contact_Method
      WHERE Entity_ID = :Entity_ID
```

where :Entity\_ID is the entity object identifier passed into this use case. Note: The Data Entry Clerk can modify the priorities in the Modify Entity use case.

15 Step 4. The Data Entry Clerk optionally enters a Description for the contact method and terminates the use case.

Step 5. The enter contact method module 1720 inserts the contact method into the Licensing Database 1204, linking the entity and contact method.

There are a number of extensions with this use case:

- 20     ° If the contact method is telephone, the Data Entry Clerk enters a country code (default 1), an area code, a telephone number, and (optionally) an extension and whether the phone is a fax line.
- ° If the contact method is email, the Data Entry Clerk enters a domain and an address.
- 25     ° If the contact method is a mail address, the Data Entry Clerk enters a street address, a city, a state or province, and a country (default = USA). Optionally, the Data Entry Clerk enters a second line of street address and/or a postal code. The Clerk can enter this data in any order.
- ° If the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Display Entities*

The display entities use case 2502 is diagramed in FIG. 25.

The Enter Entity module 1718 and Modify Entity module 1728 both use the display entities module 1722 to display organizations and people. The display entities module 1722 displays a tree of organizations and displays all the people in an organization that the Data Entry Clerk selects.

The display entities module 1722 is further described below:

Step 1. The display entities module 1722 displays a list of top level organizations in alphabetical order from the Licensing database 1204 in the organization pane 2204 of the contact view 2202. The display entities module 1722 then displays in the organization pane 2204, for each organization, the hierarchy of organizations that have the top-level organization as a parent.

Step 2. The Data Entry Clerk selects an organization from the organization pane 2204.

Step 3. The display entities module 1722 displays in the people pane 2206 an alphabetical list of all the people in the organization from the Licensing database 1204, preferably displaying the Entity unique identifier, the entity Name, the entity Description, the role Name, and the Start Date and End Date (if any) of the interval during which the person played the particular role in the organization, all preferably ordered by the last name then first name of the person. The user can select to order the data in other ways (this is true whenever data is displayed in the present invention). Note that a person may have different roles in the organization, perhaps at different times. The display entities module 1722 displays the person only once but supplies a list of roles and time periods for the person with multiple roles.

### *Modify Entity*

The modify entity use case 2602 is diagramed in FIG. 26.

The Data Entry Clerk interacts with the modify entity use case 2602 to modify entity information for people and organizations, including organization levels, people's names, and contact methods. The Data Entry Clerk may optionally modify the links between a person and their roles in organizations.

5           The modify entity module 1728 is further described below:

Step 1. The Data Entry Clerk begins the transaction by selecting an appropriate menu option, or pressing an appropriate dialog button.

Step 2. The display entities module 1722 displays a list of top level organizations in alphabetical order from the Licensing database 1204 in the organization pane 2204 of the contact view 2202. The display entities module 1722 then displays, for each organization, the hierarchy of organizations that have the top-level organization as a parent.

Step 3. The Data Entry Clerk selects an organization.

Step 4. The display entities module 1722 displays an alphabetical list of all the people in the organization from the Licensing database 1204 in the people pane 2206.

Step 5. The Data Entry Clerk selects an entity (person or organization) for modification.

Step 6. The modify entity module 1728 displays a data entry form for the appropriate kind of entity.

Step 7. The Data Entry Clerk optionally changes the entity name and/or description in any order.

Step 8. The Data Entry Clerk commits the transaction.

Step 9. The modify entity module 1728 updates the Licensing database 1204 with the modified entity information and confirms the commit to the Data Entry Clerk.

This use case has a number of extensions:

- For a person, the Data Entry Clerk modifies the individual elements of the name, which the modify entity module 1728 then concatenates together for the Entity Name.
- For an organization, the Data Entry Clerk can optionally modify an  
5 organizational level from a list.
- Optionally for an organization, the Data Entry Clerk enters or changes the date on which the organization came into existence.
- Optionally for an organization, the Data Entry Clerk links the organization to another organization as its parent (the organization the Clerk is  
10 entering is the child, the other organization is a parent). Preferably, there can be only one such parent, so the change modifies any current link.
- Optionally for a person, the Data Entry Clerk adds, modifies, or removes a role in an organization using the Link Person to Organizational Role use case, which extends this use case.
- If the Data Entry Clerk wants to enter additional contact information  
15 for the entity, the Enter Contact Method use case extends this use case. This repeats as often as necessary.
- If the Data Entry Clerk wants to change existing contact information for the entity, the Modify Contact Method use case extends this use case. This repeats  
20 as often as necessary. This use case must query the set of contact methods and let the Data Entry Clerk identify which one to modify.

### ***Modify Contact Method***

This use case is diagramed in FIG. 27.

- 25 This use case extends or is used by another use case, which supplies the object identifier for an entity and the priority of the contact method for that entity, which taken together identify the contact method. The Data Entry Clerk can change any of

the characteristics of the contact method, which may be a telephone number, a mail address, an email address, etc.

The modify contact method module 1726 is discussed further below:

5       Step 1. The modify contact method module 1726 accesses a contact method from the licensing database 1204. The modify contact method module 1726 displays the contact method in a dialog.

Step 2. The Data Entry Clerk optionally modifies the Description and details (see Extensions) for the contact method and terminates the use case.

10       Step 3. The modify contact method module 1726 inserts the contact method into the Licensing database 1204, linking the entity and contact method.

This use case includes the following extensions:

- 15       ◦ If the contact method is telephone, the Data Entry Clerk enters a country code (default 1), an area code, a telephone number, and (optionally) an extension and whether the phone is a fax line.
- If the contact method is email, the Data Entry Clerk enters a domain and an address.
- If the contact method is a mail address, the Data Entry Clerk enters a street address, a city, a state or province, and a country (default = USA). Optionally, the Data Entry Clerk enters a second line of street address and/or a postal code. The  
20       Clerk can enter this data in any order.

### ***Link Person to Organizational Role***

The link person to organization role use case 2802 is diagramed in FIG. 28.

25       This is a use case that extends other use cases when the Data Entry Clerk wants to model a particular person's role in an organization during a given period of time. The extended use case passes in the object identifier for the person and the organization. The Data Entry Clerk selects the role from a list of roles and sets the time period.

This use case is further described below:

Step 1. The extended use case passes in the object identifiers for a person and an organization.

5 Step 2. The link person to organization role module 1724 displays a form containing a list of roles ordered by name, displaying the Name and Description for each role, and entry fields for Start and End Dates.

Step 3. The Data Entry Clerk selects a role and enters the Start Date (required) and optionally the End Date.

10 Step 4. The link person to organization role module 1724 creates a link between the person, the role, and the organization in the Licensing database 1204 as an instance of Organization People and a time period using the Start and End Dates.

This use case has a number of extensions:

- ° If the Data Entry Clerk supplies only a Start Date, the link person to organization role module 1724 creates an Open-Ended Period using that date.
- 15 ° If the Data Entry Clerk supplies both a Start Date and an End Date, the link person to organization role module 1724 creates a Fixed Interval using those dates.

### *Print Entity*

The print entity use case 2902 is diagramed in FIG. 29.

20 This use case extends the Print Object use case. The print entity module 1708 prints information on an Entity, including all information in the Licensing database 1204 about the entity, its relationships to other objects, and its contact information.

The operation of this use case is further described below:

25 Step 1. The Actor via the print entity module 1708 selects an Entity and selects a command to print it.

Step 2. The print entity module 1708 prints a formatted report with the Entity ID, Name, Description, and Named Entity Type (Organization or Person).

Step 3. The print entity module 1708 prints the contact information for the entity in priority order, printing the Priority, Description, and Contact Method Type for each method.

5 Step 4. The print entity module 1708 prints the Role Name and Know How Description for any Contributor relationship the Entity participates in.

Step 5. The print entity module 1708 prints the Role Name and Agreement Number and Name for any license agreement to which the Entity is a party.

This use case has a number of extensions.

10     ° If the Entity is an Organization, the print entity module 1708 prints the parent organization's Name and the Description of the Organizational Level of the organization.

° If the Entity is a Person, the print entity module 1708 prints the Organizations to which the Person belongs in order by time period.

15     ° If the Contact Method is a Telephone, the print entity module 1708 prints the number in the format "+<Country Telephone Code > <Area Code> <Telephone Number> x<Extension>." If Is Fax is true, the print entity module 1708 adds "(facsimile)" to the string.

20     ° If the Contact Method is an Address, the print entity module 1708 prints the address in the format "<Address 1>, <Address 2>, <City>, <State or Province>, <Country Abbreviation>, <Postal Code>".

° If the Contact Method is an Email, the print entity module 1708 prints the address in the format "<Address>@<Domain>".

### ***Remove Entity***

25     The remove entity use case 12102 is diagramed in FIG. 121. According to this use case, the Licensing system Administrator selects an entity and removes it from the Licensing database 1204. Preferably, an entity can be removed only if it does not participate as a party to a license agreement or as a contributor to a



know-how asset. In removing an entity, a number of different objects are also removed:

- The entity's contact methods
- Organization or person corresponding to the entity
- 5      ◦ Links from the entity to an organization/role combination
- Name fragments associated with the person

The operation of this use case is further described below.

Step 1. The License Administrator begins the transaction.

10      Step 2. The Remove entity module 1794 displays a list of existing entities ordered by name, displaying the name and entity type.

Step 3. The License Administrator selects an entity and removes it. This is allowed only if the entity is not associated with a license (no rows in the Party table for this entity) or has no payments (no rows in the Payment table with Payor\_ID = OID) . The License Administrator can select and remove multiple entities at once or  
15      in sequence. At the end of the process, the License Administrator commits the transaction.

Step 4. The Remove entity module 1794 deletes the entity, removing also the corresponding subclass (Person or Organization), links to Organization for people, name fragments, and contact methods.

20      Step 5. The Remove entity module 1794 confirms the commit to the License Administrator.

This use case has a number of extensions:

- If there are any payments linked to the entity, the Remove entity module 1794 disallows removal and reports the error message " <Name> has one or  
25      more payments in the remove entity module 1794, so you cannot remove it." The <Name> is the name of the entity.

- If there are any licenses linked to the entity, the Remove entity module 1794 disallows removal and reports the error message " <Name> has one or more

license agreements in the remove entity module 1794, so you cannot remove it." The <Name> is the name of the entity.

° If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Context-Sensitive Help use case.

5

### *Query Entities*

The query entities use case 15402 is diagramed in FIG. 154. This use case is used by the Link Payment to Entity use case to supply an entity to that use case for linking to a license payment. The License Administrator (and potentially any User) queries a named entity from the Licensing Database 1204 based on any of the structured fields of the entity (name, description, and entity type). The query entities module 15404 displays the name, description, and entity type of the named entities that match the query conditions and allows the License Administrator to select some for further processing in a using use case. The operation of this use case is further described below.

15

Step 1. The query entities module 15404 displays a query form with the name, description, and entity type fields available for specifying a query.

Step 2. The License Administrator enters the search criteria.

Step 3. The query entities module 15404 queries all the entities that meet the query criteria from the Licensing Database 1204. The query entities module 15404 displays the entities, showing the name, description, and type.

20

Step 4. The License Administrator selects one or more entities for use in the using use case.

### *Asset Use Cases*

25

An asset is an intellectual property document (for example, patent, copyright, trademark, trade secret, etc.) or an intellectual asset (for example, know how).

According to the invention, assets are combined into asset packages, which may be standard (lists of assets), group (reference to an IPAM group, for example, described above), or descriptive (a text describing the assets with language from the license agreement).

5            Preferably, a group as described above (sometimes herein refer to as an IPAM group) differs from an asset package. When the licensing system 1102 is integrated with the IPAM server 214, it is possible to create a type of an asset package (the IPAM group type) that links to a group and that acquires all of the documents associated with that group. In an embodiment, changes to the contents of the group automatically  
10           affect the asset package (in other embodiments, changes to the contents of the group do not automatically affect the asset package). The asset package is said to be linked (where the link is said to be "active") or associated with the group. But the asset package itself is a separate object from that group.

             Linking the package to a license agreement preferably means that the license  
15           licenses all of the assets in the package to the licensee, unless the license agreement indicates otherwise.

             An asset view 3002 is displayed when the asset packages icon 1416 is pressed (FIG. 30). The Asset View 3002 contains two panes. The first pane, called the asset packages pane 3004, displays a spreadsheet of asset packages, and the second pane,  
20           called the assets pane 3006, displays the assets in the asset package selected in the asset packages pane 3004.

             Selecting an asset package in the asset packages pane 3004 displays the assets in that package in the assets pane 3006. Double-clicking on an asset package in the asset packages pane 3004 displays the asset package modification dialog, while  
25           double-clicking on the asset in the asset pane 3006 displays the asset modification dialog. You can enter a new package or asset by double-clicking on an empty record or clicking on the New button. If no package is selected, you create the new asset independently of any package; otherwise, you create the asset in the selected package.

Clicking on a Find tool displays the Find Asset dialog (FIGS. 64A and 65, for example), which lets you enter search conditions to search for an asset. The find tool may be accessed in a number of ways, such as from the tool bar or via the menu (see FIG. 67). Entering search conditions into this dialog and clicking on Find Now  
5 displays the matching assets in the extended dialog. See FIGS. 64B and 66. You can then double-click on the asset to display the modification dialog with all the details of the asset, or you can drag-and-drop or copy-and-paste the assets into a selected asset package.

### *Enter Patent*

10 The enter patent use case 3102 is diagramed in FIG. 31.

The enter patent use case 3102 enables a Data Entry Clerk to enter information for a patent (U.S. or foreign) asset. The patent can be packaged alone or with other assets into asset packages that can be licensed.

The operation of the enter patent use case 3102 is further described below:

15 Step 1. The Data Entry Clerk starts a transaction to enter a new patent asset by issuing an appropriate command. This can be done, for example, by selecting New, Patent from the menu bar 1406 (FIG. 32).

Step 2. The enter patent module 1730 displays a data entry form called an enter new patent dialog 3302 (FIG. 33) for input and generates an object identifier for  
20 the new asset. The enter new patent dialog 3302 has two tabs, a general tab 3304 and a optional tab 3306.

Step 3. The Data Entry Clerk chooses from a list of publishing organizations and document kinds. For example, the Data Entry Clerk can press a down arrow key 3308 (FIG. 33) to view a list of available publishing organizations 3402 (FIG. 34).  
25 The Data Entry Clerk can press another down arrow key 3312 to view and choose from a list of different types of documents, such as utility patent, design patent, plant patent, SIR, provisional application, patent application, etc.

Step 4. The Data Entry Clerk enters data into the following fields in any order: Document Number, Long Display Name, Patent Type, and Filing Date. Optionally, the user can enter the following fields in any order: Description, Title, Filing Date, Issue Date, and Publication Date. See FIGS. 33 and 35. The Data Entry Clerk sets the Security Class from a list of available security classes with Write privilege for the user.

Step 5. The Data Entry Clerk commits the transaction by pressing a save button 3310 (FIG. 33).

Step 6. The enter patent module 1730 creates an IP Document, a Document, and an object of the appropriate subclass for the patent type in the Core Database 1208. The enter patent module 1730 marks the Document as an IP Document.

This use case has a number of extensions:

- ° If the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

- ° If the IPAM Server 214 and IPAM database 216 are available, the Licensing System 1102 disallows entry of patent information via the enter patent module 1730. Essentially, the enter patent module 1730 cannot be accessed when the IPAM Server 214 and IPAM database 216 are available. Instead, all patent information is obtained from the IPAM Server 214 and IPAM database 216.

### ***Enter Trademark***

The enter trademark use case 3602 is diagrammed in FIG. 36.

The Data Entry Clerk enters information for a trademark asset, including the trademark classes and the kind of trademark (US Federal, State, WIPO, etc.).

The operation of the enter trademark use case 3602 is further described below.

Step 1. The Data Entry Clerk starts a transaction to enter a new trademark asset. This can be done in a number of ways, such as by selecting New, Trademark from the menu bar 1406 (FIG. 32).

Step 2. The enter trademark module 1732 displays a data entry form 3702 (FIG. 37) for input of trademark information and generates an object identifier for the new asset. The data entry form 3702 has a general tab 3704, an optional tab 3706, and a classes tab 3708.

5           Step 3. The Data Entry Clerk chooses from a list 3802 of publishing organizations (FIG. 38) and document/trademark kinds (by pressing a down arrow key 3712 to display a list of document/trademark kinds, where such document/trademark kinds are any known types).

10           Step 4. The Data Entry Clerk enters data into the following fields in any order: Doc Kind, Document Number, Long Display Name, and Filing Date. Optionally, the user can enter the following fields in any order: Description, Title, Filing Date, Issue Date, and Publication Date. The Data Entry Clerk enters a set of Trademark Classes from a list of classes. The Data Entry Clerk sets the Security Class from a list of available security classes with Write privilege for the user. See FIGS.  
15           37-40.

          Step 5. The Data Entry Clerk commits the transaction by pressing a save button 3804.

20           Step 6. The enter trademark module 1732 creates an IP Document, a Document, an object of class Trademark, and an association object relating the trademark to a series of Trademark Classes in the Licensing Database 1204. The enter trademark module 1732 marks the Document as an IP Document.

          At any time, if the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### ***Enter Copyright***

25           The enter copyright use case 4102 is diagramed in FIG. 41. In this use case, the Data Entry Clerk enters basic information for a copyright asset.

          The operation of the enter copyright use case 4102 is further described below:

Step 1. The Data Entry Clerk starts a transaction to enter a new copyright asset. This can be done in a number of ways, such as by selecting New, Copyright from the menu bar 1406 (FIG. 32).

5 Step 2. The enter copyright module 1734 displays a data entry form 4202 (FIG. 42) for input and generates an object identifier for the new asset. The data entry form 4202 has a general tab 4206 and an optional tab 4208.

Step 3. The Data Entry Clerk chooses from a list of publishing organizations and document kinds. See, for example, FIG. 43.

10 Step 4. The Data Entry Clerk enters data into the following fields in any order: Document Number, Long Display Name, and Filing Date. Optionally, the user can enter the following fields in any order: Description, Title, Filing Date, Issue Date, and Publication Date. The Data Entry Clerk sets the Security Class from a list of available security classes with Write privilege for the user. See FIGS. 42-44.

15 Step 5. The Data Entry Clerk commits the transaction by pressing a save button 4204.

Step 6. The enter copyright module 1734 creates an IP Document, a Document, and an object of class Copyright in the Licensing Database 1204. The enter copyright module 1734 marks the Document as an IP Document.

20 At any time, if the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### ***Enter Trade Secret***

The enter trade secret use case 4502 is diagramed in FIG. 45. According to this use case, the Data Entry Clerk enters information for a Trade Secret asset.

25 The operation of the enter trade secret use case 4502 is further described below:

Step 1. The Data Entry Clerk starts a transaction to enter a new trade secret asset using any procedure described herein.

Step 2. The enter trade secret module 1736 displays a data entry form for input and generates an object identifier for the new asset.

Step 3. The Data Entry Clerk chooses from a list of publishing organizations and document kinds.

5           Step 4. The Data Entry Clerk enters data into the following fields in any order: Document Number, Long Display Name, and Filing Date. Optionally, the user can enter the following fields in any order: Description, Title, Filing Date, Issue Date, and Publication Date. The Data Entry Clerk sets the Security Class from a list of available security classes with Write privilege for the user.

10           Step 5. The Data Entry Clerk commits the transaction.

Step 6. The enter trade secret module 1736 creates an IP Document, a Document, and an object of class Trade Secret in the Licensing Database 1204. The enter trade secret module 1736 marks the Document as a type of know how.

15           As always, at any time, if the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Context-Sensitive Help use case.

### ***Enter Know How***

The enter know how use case 4602 is diagramed in FIG. 46. In this use case, the Data Entry Clerk enters information for a Know-How asset into the Licensing Database 1204.

20           The operation of this use case is further described below:

Step 1. The Data Entry Clerk starts a transaction to enter a new know how asset. This can be done in a number of ways, such as by selecting New, Know How from the menu bar 1406 (FIG. 32).

25           Step 2. The enter know how module 1738 displays a data entry form 4702 (FIG. 223) for input and generates an object identifier for the new asset.

Step 3. The Data Entry Clerk enters various items:



Step 3.1. The Data Entry Clerk sets the Description of the know how by entering text. The Data Entry Clerk can also link this asset to one or more other objects (such as documents, figures, software programs, etc.) that represent or illustrate the know how asset.

5 Step 3.2. The Data Entry Clerk chooses from a list of intellectual asset kinds. This list can include any standard or use defined terms describing intellectual asset kinds, and could be based on technology area, market area, industrial area, corporate organization level, etc.

10 Step 3.3. The Data Entry Clerk sets the Security Class from a list of available security classes with Write privilege for the user. The security class indicates the users who have access to this record.

Step 4. The Data Entry Clerk commits the transaction.

15 Step 5. The enter know how module 1738 creates a Document and an object of class Know How in the Licensing Database 1204. The enter know how module 1738 marks the Document as a Know How.

At any time, if the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Modify IP Asset*

20 The modify IP asset use case 4802 is diagramed in FIG. 48. According to this use case, the License Administrator queries an IP asset and modifies it. The operation of this use case is further described below.

Step 1. The License Administrator begins the transaction to modify an IP asset using any of the techniques described herein.

25 Step 2. The modify IP asset module 1740 uses the Query Assets module 1742 to display a set of assets and to allow the License Administrator to select one for modification.

Step 3. If the License Administrator has Write permission on the selected asset, the IP asset module 1740 displays a form containing, for example, these fields for the asset document: Document Number, Title, Long Display Name, Issue Date, Filing Date, Publication Date, Description, Asset Type ("Disc Switch"), Publishing Organization (Description from Pub Organization class), and Doc Kind (Description from IP Doc Kind class).

Step 4. The License Administrator modifies any of the fields in any order and signals the end of the transaction.

Step 5. The IP asset module 1740 writes the changes to the licensing database 1204 or the core database 1208 and confirms the commit to the License Administrator.

There are a number of extensions of this use case:

- If the asset is a patent from the IPAM database 216, the user cannot modify any of the asset details.
- If the asset is a patent (JPO\_PATENT, EPO\_PATENT, PCT, or PTO\_PATENT classes), the extended fields include Assignee, Class, International Class, and Inventor.
- If the asset is a Trademark, the extended fields include Trademark Class Display Name and Description.
- If the asset is a Know How, the extended fields include the IA Kind Description and the Know How Description.
- If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Asset Package Use Cases*

An asset package includes one or more IP assets. Use cases related to asset packages are described below.

### *Create IP Asset Package*

The create IP asset package use case 4902 is diagramed in FIG. 49. According to this use case, the License Administrator creates a named package of intellectual property (IP) assets in the Licensing Database 1204. All or part of this asset package (possibly along with one or more other asset packages) may then be licensed via a license agreement. An asset package can be generated in a number of ways: by querying the assets from the Licensing Database 1204, by querying a group from the Core Database 1208, or by entering a textual definition of the assets. The operation of this use case is further described below.

Step 1. The License Administrator begins the transaction to create a new IP asset package. This can be done in a number of ways, such as by selecting New, Asset Package from the menu bar 1406 (FIG. 32).

Step 2. The create IP asset package module 1744 displays an asset package dialog 5002 (FIG. 50). From a pull down list 5004, the License Administrator chooses the type of asset package to create (Asset/Standard, Group, or Descriptive Package Type). The Asset/Standard type is an asset package containing any combination of assets selected by the user. The Group type is an asset package containing the assets included in one or more pre-existing groups. The Descriptive Package type is an asset package whose assets are indicated by way of a description, such as a textual description, an audio description, a multimedia description (the files for those media types may be linked to this record).

Step 3. The create IP asset package module 1744 creates a new package with a new object identifier.

Step 4. The License Administrator enters a Name for the package. Optionally, and in any order, the License Administrator enters the Description and/or the capture period (Start and End timestamps). See FIG. 51.

If the Asset/Standard type is selected, then an asset/standard type dialog 5102 is displayed (FIG. 51). The asset/standard type dialog 5102 has a general tab 5104,

assets tab 5106, and allocation tab 5108. The License Administrator can use the Query Assets module 1742 to query a set of asset documents from the Licensing Database 1204 for the package. The create IP asset package module 1744 links these documents to the new package. The new assets are listed in the assets tab 5106 (FIG. 52).

5

If the Package Type is Asset/Standard or Group, the License Administrator can choose to allocate the assets within the package. The create IP asset package module 1744 displays a list of the assets in the package, and the License Administrator enters an Allocation Percent for each asset. See FIG. 53. The License Administrator can select an even distribution or a custom distribution, where relative percentages are entered by the License Administrator. These allocations affect the manner in which license fees will be attributed to the assets in a package. For example, suppose an asset package includes an Asset1 having an allocation of 60%, and an Asset2 having an allocation of 40% (as shown in the example of FIG. 53). In this case, 60% of any license revenue attributed to this asset package will be considered to have been generated by Asset1, and 40% of any license revenue attributed to this asset package will be considered to have been generated by Asset2.

10

15

If the Group type is selected, a group dialog 5402 is displayed (FIG. 54) having a general tab 5104, a group tab 23004, and an allocation tab 5108. The create IP asset package module 1744 displays a list of group names from the Core Database 1208, preferably in alphabetical order. The License Administrator as a User must have the ability to read the groups in the list. The License Administrator selects one or more groups that hold the documents/assets that the package is to package. The create IP asset package module 1744 displays the group documents and links them to the package in the group tab 5404 (FIG. 55).

20

25

If the Descriptive type is selected, then a descriptive dialog 5602 is displayed (FIG. 56) having a general tab 5104 and a description tab 5604. The License Administrator enters text regarding Inclusion Criteria that describes the legal qualities

of the assets, preferably in the language from the Licensing Agreement, in addition to the Standard behavior. See FIG. 57.

5 Step 5. The create IP asset package module 1744 writes the new asset package to the Licensing Database 1204, making the License Administrator user the owner of the package, then confirms the transaction commit to the License Administrator.

### *Modify IP Asset Package*

10 The modify IP asset package use case 5802 is diagramed in FIG. 58. According to this use case, the License Administrator selects an IP Asset Package from the Licensing Database 1204, then modifies the information about the package and/or its contents: the documents listed in a standard package, the group contained within a Group package, or the text contained within a Descriptive package. The operation of this use case is further described below.

15 Step 1. The License Administrator starts the transaction to edit an IP asset package. This can be done, for example, by selecting the asset packages icon 1416 to display an asset package view 5902. The asset package view 5902 has a package list pane 5904 where asset packages are listed, and an asset list pane 5906 where assets in a selected asset package (selected in the package list pane 5904) are listed.

20 Step 2. The modify IP asset package module 1746 displays the asset packages in the packages list pane 5904 by querying the Licensing Database 1204, preferably displaying the Name, Description, and/or Package Type preferably in alphabetical order by name. The modify IP asset package module 1746 displays only those packages for which the License Administrator has Read permission or ownership.

25 Step 3. The License Administrator selects a package in the packages list pane 5904.

Step 4. If the License Administrator has Write permission on the selected package or owns it, the modify IP asset package module 1746 displays a form

showing all the fields of the package, including the object identifier, the Package Name, the Description, the capture period (Start and End date/time), and the Package Type. This is similar to the examples shown in FIGS. 50-57.

5 Step 5. The License Administrator changes the Name, Description, Capture Period, Package Type, or other information in any order, then ends the transaction.

Step 6. The modify IP asset package module 1746 updates the Licensing Database 1204 with the changes and confirms the commit to the License Administrator.

This use case has a number of extensions:

10       ° If the License Administrator changes Package Type, the modify IP asset package module 1746 uses the same logic as in the Create IP Asset Package use case 4902 to create a new subclass for the package with the appropriate contents. The modify IP asset package module 1746 removes any asset allocation percents. The modify IP asset package module 1746 can undo the change until the License Administrator ends the transaction.

15       ° If the Package is a Standard package, the modify IP asset package module 1746 displays a list of assets. The License Administrator can add or remove assets from the list. The License Administrator uses the Query Assets use case to query a set of additional asset documents from the Core Database 1208 for the package. The modify IP asset package module 1746 links these documents to the new package. The License Administrator selects and removes asset documents. The License Administrator adjusts the Allocation Percent of each asset as required.

20       ° If the Package is a Group package, the modify IP asset package module 1746 displays the Name and Description of the group. The License Administrator can change the group by selecting from a list of available groups (groups from the Core database 1208 that the user has at least read access to, see the query in the Create IP Asset Package use case). The modify IP asset package module 1746 displays a list of group names from the Core Database 1208 in alphabetical order. The License Administrator as a User must have the ability to read the groups

25

in the list. The License Administrator selects a group that holds the documents that the package packages. The modify IP asset package module 1746 displays the group documents and links them to the package. The modify IP asset package module 1746 removes current Allocation Percent values (all links get deleted).

5           °       If the Package is a Descriptive package, the modify IP asset package module 1746 displays the Description. The License Administrator can change the Description by editing the text.

              °       If the License Administrator wants to modify the allocation of assets within the package, the modify IP asset package module 1746 displays the current  
10       allocations and the Administrator changes them..

### *Print Asset Package*

The print asset package use case 6002 is diagramed in FIG. 60. This use case extends the Print Object use case, and the Print License use case uses it. The print asset package module 1710 prints an Asset Package, including all information in the  
15       Licensing Database 1204 and the Core Database 1208 about the package and its documents. The operation of the print asset package module 1710 is further described below.

Step 1. The Actor in the Print Object use case or the Print License Use case selects an Asset Package and prints it.

20           Step 2. The print asset package module 1710 prints a formatted report with the Start Date, End Date, Name, Description, Package Type, etc. It then prints package-specific information (see Extensions).

Step 3. The print asset package module 1710 prints the Document IDs of the documents and the document-specific information (see Extensions).

25           This use case has a number of extensions:

- If the package is a Descriptive Asset Package, the print asset package module 1710 prints the Inclusion Criteria for the package and does not print any documents.
- If the package is a Group Asset Package, the print asset package module 1710 prints the Group ID, Name, and Description of the group and prints the IP Documents and UDDs in the group.
- If the package is a Standard Package, the print asset package module 1710 prints the Documents in the package.
- If a Document is an IP Document, the print asset package module 1710 prints the Publishing Organization Description, the IP Doc Kind Description, the Document Number, the Title, and the Description.
- If a Document is a Know-How, the print asset package module 1710 prints the Intellectual Asset Kind Description and the Description.
- If a Document is a User Defined Document (UDD), the print asset package module 1710 prints information pertaining to the document.

### ***Remove IP Asset***

The remove IP asset use case 6102 is diagramed in FIG. 61. This is preferably an administrator use case. According to this use case, the licensing system administrator removes an IP asset (an IP Document or a Know How document, for example) and its related data from the Licensing Database 1204 and Core Database 1208. The operation of this use case is further described below:

Step 1. The System administrator begins the transaction.

Step 2. The remove IP asset module 1703 displays a list of all assets (the IP Documents and Know How Documents) in the Licensing Database 1204 and Core Database 1208, displaying the Document ID, the Document Type (disc\_switch for IP Documents and IA\_Kind Description for Know How), the Document Number and Title for IP Documents, ordering by the type and document ID.



Step 3. The Administrator selects one or more assets and removes them..

Step 4. The remove IP asset module 1703 deletes the information from the Licensing and Core databases 1204 and 1208 and moves it to the Recycle Bin (implementation may differ, deferring the actual SQL deletes, and there is no notification of commit to the Administrator).

This use case has a number of extensions:

- ° At any time until clearing the Recycle Bin, the administrator can restore the removed Asset.

- ° If the system integrates with the IPAM Server 214, the Administrator sees only assets from the Licensing Database 1204, not those from the Core 1208 (JPO\_Patent, EPO\_Patent, PCT, and PTO\_Patent). The deletes from the IP\_Document and Document classes are done with Administrator privileges in the Core database 1208 but do not affect patents in any way.

- ° If the Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### ***Remove IP Asset Package***

The remove IP asset package use case 6202 is diagrammed in FIG. 62. According to this use case, the License Administrator selects an asset package from a list and removes it from the Licensing Database 1204, subject to security access and links to license agreements. The operation of this use case is further described below.

Step 1. The License Administrator begins a transaction.

Step 2. The remove IP asset package module 1748 displays a list of the asset packages to which the user has Read access or which the user owns, displaying the object identifier, the Name, the Description, and the Package Type, ordered by Name.

Step 3. The License Administrator selects one or more asset packages to which the user has Delete access or which the user owns and removes them. The License Administrator then commits the transaction.

Step 4. The remove IP asset package module 1748 removes the persistent asset packages from the Licensing Database 1204 and confirms the operation to the License Administrator. The remove IP asset package module 1748 propagates the delete to the appropriate subclass based on Package\_Type (Standard Asset Package, Descriptive Asset Package, Group Asset Package) and to the Secure Object superclass.

This use case has a number of extensions.

- If the asset package has links to a license agreement, the remove IP asset package module 1748 does not allow removal, displaying the error message "The <oid> license agreement uses this asset package; you cannot remove it." The <oid> is the object identifier for the license agreement.

- If the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Context-Sensitive Help use case.

### *Query Assets*

The query assets use case 6302 is diagramed in FIG. 63. This use case is used by other use cases. The License Administrator queries the Licensing Database 1204 for a set of asset documents by entering search criteria into a form (see FIGS. 64-66, for example). The License Administrator can remove assets from this set individually. The use case returns a set of License object identifiers (OIDs). The operation of this use case is further described below.

Step 1. The using use case calls this use case, and the query assets module 1742 displays, for example, a screen that contains these fields for search criteria: Document Number, Title, Long Display Name, Issue Date, Filing Date, Publication Date, Description, Asset Type ("Disc Switch"), Publishing Organization (Description from Pub Organization class), Doc Kind (Description from IP Doc Kind class), and/or any other search fields discussed herein, or conventionally used. See example displays in FIGS. 64A, 64B, and 65.

Step 2. The License Administrator enters one or more fields of search criteria and submits the query.

Step 3. The query assets module 1742 displays a list of documents 6602 that satisfy the query, displaying the Asset Type, Document Number, and Title (for IP documents) and/or IA Kind and Description (for Know How), and to which the License Administrator has READ access. See FIGS. 64B and 66.

Step 4. The License Administrator may take any number of actions, including one of these actions:

Step 4.1. The License Administrator may select an asset for editing (Modify IP Asset use case).

Step 4.2. The License Administrator may drag and drop or copy and paste an asset into a list of assets (Create or Modify IP Asset Package).

Step 5. The License Administrator ends the use case.

This use case has a number of extensions:

° If the asset is a patent (JPO\_PATENT, EPO\_PATENT, PCT, or PTO\_PATENT classes), the extended search criteria includes Assignee, Class, International Class, and Inventor.

° If the asset is a Trademark, the extended search criteria includes Trademark Class Display Name and Description and the Trademark Kind Description.

° If the asset is a Know How or Trade Secret, the extended search criteria includes the IA Kind Description and the Know How Description.

° If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

## ***Query Asset Packages***

The query asset packages use case 15902 is diagramed in FIG. 159. According to this use case, the Auditor, the Data Entry Clerk, or the License

Administrator (and potentially any actor) queries an asset package from the Licensing Database 1204 based on any of the structured fields of the package (including inclusion criteria for descriptive asset packages and group name and description for group packages) or the list of assets (documents) associated with the package. The query asset package module 33504 displays the set of asset packages that qualify given the search criteria. The Modify Asset Package use case extends this one, enabling the actor to modify a package found through the query. This use case is further described below.

Step 1. The Actor begins the transaction.

Step 2. The query asset package module 15904 displays a form that allows the Actor to specify a set of search criteria for the query. The main part of the form contains these fields:

- \* Start Date (date on which asset package becomes effective, start of availability of assets in the package)

- \* End Date (date on which asset package terminates, end of availability of assets in the package usually due to patent expiration or similar asset loss)

- \* As Of Date (a date that is potentially between the Start Date and End Date, allowing the actor to search for asset packages that are effective as of the date)

- \* Package Name (the text name of the package)

- \* Package Description (the text describing the nature of the package)

- \* Package Type (Standard, Descriptive, or Asset), defaulting to Standard

Step 3. The Actor enters any of the fields and tells the query asset package module 15904 to execute the query.

Step 4. The query asset package module 15904 concatenates all the conditions with a logical AND, then queries the Licensing Database 1204 using the resulting search condition. The query asset package module 15904 then displays a list of all the

asset packages that satisfy the search condition. The query asset package module 15904 displays the Package Name, Description, and Package Type.

Step 5. The Actor ends the transaction.

This use case has a number of extensions:

- 5           °       If the Actor selects a Package Type of Standard or Descriptive, the query asset package module 15904 displays an additional form that allows the user to enter a search expression that specifies a logical conditional expression of ANDs, ORs, and NOTs (with parenthetical grouping) combining assets ((Asset 1 AND Asset 2) OR Asset 3) which it gathers with the Query Assets use case. The query asset  
10       package module 15904 queries the Licensing Database to find those packages with the appropriate combination of assets.
- °       If the Actor selects a Package Type of Descriptive, the query asset package module 15904 displays an additional form that allows the user to enter a text search on Inclusion Criteria. The query asset package module 15904 queries the  
15       Licensing Database to find those Descriptive packages that qualify.
- °       If the Actor selects a Package Type of Group, the query asset package module 15904 displays an additional form that allows the user to enter a text search on Group Name and/or Group Description. As well, the query asset package module 15904 displays a form for entering a search expression for grouped documents with  
20       AND, OR, and NOT relational operators as for Standard or Descriptive packages. The query asset package module 15904 queries the Licensing Database to find those group packages with the appropriate combination of assets.

### *License Agreements Use Cases*

- 25       A license agreement is a contract in which an entity, the licensor, licenses intellectual property to another entity, the licensee. There may be a plurality of licensors and/or licensees. In addition to a variety of contract clauses and the links to any number of asset packages, the license agreement in the context of the present

invention contains definitions of the parties to the agreement, compensation terms, territorial restrictions, field-of-use (market area) restrictions, and any other terms and/or clauses used in license agreements.

5 A License Agreements View 6802 (FIG. 68) displays a spreadsheet pane of all the license agreements. Double-clicking on a license displays the agreement modification dialog 6902 (FIG. 69). This dialog 6902 displays five tabs. The General tab 6904 shown here contains the basic information about the agreement. The Asset Packages tab 6906 lists the asset packages that are the subject of the license and lets the License Administrator administer the list. The Terms tab 6908 lists the  
10 compensation terms of the license (fees, royalties, advances, minimum guarantees, and other). The License Administrator can modify these terms through that tab. The Royalty Statements tab 6910 and Payments tabs 6912 let the License Administrator see the linked statements and payments related to the license, but not modify them.

15 Double-clicking on the empty row or on the new button in the license agreements view 6802 displays the license creation dialog 7202 (FIG. 72). This dialog is the same as the modification dialog but does not display the Statements and Payments tabs, as a new license has none.

20 Selecting a Find tool in this view displays the Find Agreement dialog 7002 (FIG. 70), which lets you enter search terms for agreements. Entering search conditions into this dialog 7002 and clicking on Find Now displays the matching agreements in the extended dialog 7102 (FIG. 71). You can then double-click on the agreement to display the modification dialog with all the details of the license.

### ***Enter License Agreement***

25 The enter license agreement use case 7302 is diagrammed in FIG. 73. According to this use case, the Data Entry Clerk enters a license agreement into the Licensing Database 1204. Entering the agreement entails entering structured data (name, description, effective date, and so on). The Data Entry Clerk enters a set of

compensation terms that describe the fee and royalty structure of the agreement. The Data Entry Clerk also links the agreement to a set of asset packages representing the licensed IP assets and to the various parties to the agreement (licensor, licensees, attorneys, and so on). The operation of this use case is further described below.

5                   Step 1. The Data Entry Clerk begins the transaction.

Step 2. The enter license agreement module 1750 generates a new object identifier for the license agreement and displays a data entry form 7202 for the new license.

10                   Step 3. The Data Entry Clerk enters data into the structured data fields of the license creation dialog 7202 and links other objects as required:

Step 3.1.           The Data Entry Clerk enters the structured fields of the agreement: the Agreement ID, the Title/Name, Description, Effective Date, Expiration Date, Exclusivity, Assignable, Transferable, Revocable, Confidential, Terminated, etc.

15                   Step 3.2.           The Data Entry Clerk enters the structured fields of the license agreement: Perpetual, Infringement Based (see Extensions), Can Sublicense, Annual Audit, Audit at Licensee Expense, Favored Nation, Improvements, Grant Back, Withholding Percentage, etc.

20                   Step 3.3.           The Data Entry Clerk uses the Link to Asset Packages use case to specify the set of asset packages for the license. These linked asset packages are displayed in the asset packages tab 6906.

25                   Step 3.4.           The Data Entry Clerk uses the Enter Compensation Term use case to enter the compensation terms for the license. This step repeats until the clerk enters all the terms in the license agreement. These terms are displayed in the terms tab 6908.

Step 3.5.           The Data Entry Clerk uses the Link to Party use case to specify a party to the agreement. Required parties include one licensor and at least one licensee. You can also add other parties such as attorneys and other roles entered through the Add Role use case by the License System Administrator. This step repeats

until the clerk has linked all the parties to the agreement. This information is displayed in the Licensor and Licensee fields (FIG. 72).

Step 3.6. The Data Entry Clerk selects the Security Class for the new document from a list of available security classes, which control who has access to this license.

Step 4. The Data Entry Clerk commits the transaction.

Step 5. The enter license agreement module 1750 creates the License Agreement or Infringement Based Agreement in the Licensing Database 1204. The enter license agreement module 1750 confirms the commit to the Data Entry Clerk.

This use case has a number of extensions:

- If the correct asset package or packages don't exist, extend the use case with the Create IP Asset Package use case to create the appropriate package(s).
- If the correct entity does not exist, extend the use case with the Enter Entity use case to create the appropriate entity.
- If the Data Entry Clerk indicates that the license agreement is infringement based, create an Infr Based Agreement rather than a License Agreement and add the Infringement Release Granted flag and the Infringement Comment field to the data entry form. A license is infringement based if it resulted or is somehow associated with or related to an allegation of infringement (whether or not an infringement suit was filed in court). An embodiment of the invention supports additional information that you track for infringement based license agreements.
- If the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

#### *Link to Party*

The link to party use case 7402 is diagramed in FIG. 74. According to this use case, a Data Entry Clerk selects an entity from a list and a role from another list.



The Licensing Database 1204 creates a link between the entity, the role, and the license agreement from the using use case. This use case is further described below.

Step 1. The Data Entry Clerk starts the use case with an open transaction containing a license agreement that will link to the entity and role, passing in the object identifier for the license agreement.

Step 2. The link to party module 1756 queries the Licensing database 1204 and displays the names and descriptions of the entities and roles.

Step 3. The Data Entry Clerk selects one entity from the entity list and one role from the role list.

Step 4. The Data Entry Clerk signals the end of the use case.

Step 5. The Licensing Database 1204 creates a Party link using the License Agreement OID, the Entity OID, and the Role OID.

This use case has a number of extensions.

- ° The user can cancel the use case at any time. The result will be that the license agreement is unchanged from its status before using the use case, with no additional Party links.

- ° If the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Link to Asset Package*

The link to asset package use case 7502 is diagramed in FIG. 75. According to this use case, a Data Entry Clerk selects one or more asset packages from a list. The link to asset package module 1752 creates a link between the input license agreement and the selected asset packages. The Data Entry Clerk sets the territorial and field-of-use restrictions (and any other license details) on each asset package link. The Data Entry Clerk indicates whether the link is a cross license and/or has any further limitations, obligations, rights, etc. as part of this license. The link to asset package module 1752 is further described below.

Step 1. The Data Entry Clerk starts the use case with an open transaction containing a license agreement that will link to the asset packages. The extended use case supplies the object identifier for the agreement.

5 Step 2. The link to asset package module 1752 displays the names and descriptions of the asset packages in a list ordered by name.

Step 3. The Data Entry Clerk selects one or more packages from the list.

10 Step 4. The link to asset package module 1752 creates and displays a link between the license agreement and each package the Data Entry Clerk selects and assigns default territorial and field-of-use values. The link to asset package module 1752 sets the default Cross-Licensed flag to false.

15 Step 5. The Data Entry Clerk adds territorial and field-of-use restrictions to the links using lists of the available territories and fields of use from the Licensing Database 1204. The Data Entry Clerk also sets the Cross-License flag to true if the particular package represents assets licensed by the licensee to the licensor instead of vice versa. The Data Entry Clerk removes unwanted territorial and field-of-use restrictions from the links. The Data Entry Clerk enters any additional limitations in the Limitations field. All of these actions may occur in any order. The Data Entry Clerk signals the end of the use case.

20 Step 6. The Licensing Database 1204 links the asset packages and the input license agreement and adds links to the required territories and fields of use for each link.

This use case has a number of extensions:

- 25     ° The user can cancel the use case at any time. The result will be that the license agreement is unchanged from its status before using the use case.
- ° If the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Enter Compensation Term*

The enter compensation term use case 7602 is diagramed in FIG. 76.

5       The Enter License Agreement and Modify License Agreement use cases use  
this one to enter compensation terms to a license agreement in the Licensing Database  
1204. The Licensing Administrator or Data Entry Clerk creates a licensing term with  
the details appropriate to the particular term. All terms have a description, an amount,  
and late payment penalties, and perhaps other attributes. Ongoing royalties have a  
time period and possibly tables of escalating amounts based on number of units/sales  
plus tables of estimated number of units/sales. Ongoing royalties break down into  
10       royalties on units sold, on revenue, or on revenue from sublicense royalties. A  
minimum guarantee also has a time period and possible escalations. An advance has  
a due date and may be refundable or not. A fee may be ongoing with a time period or  
a lump sum fee with a due date.

The operation of this use case is further described below.

15       Step 1. The using use case passes in the object identifier for the license  
agreement to which the Data Entry Clerk or License Administrator (the Actor) wants  
to add compensation terms.

20       Step 2. The enter compensation term module 1754 displays a list of the  
current set of compensation terms (if any) for this license, preferably in order of term  
number, displaying the term Description, the compounding period Description, the  
Amount with its currency symbol, the Late Payment Interest Rate (if any), and the  
Term Type.

25       Step 3. The Actor creates a new term and selects one of the term types  
(Royalty Per Unit, Royalty Per Sales Amount, Royalty Per Royalty Amount,  
Minimum Guarantee, Advance, Ongoing Fee, Lump Sum Fee, or Other).

Step 4. The enter compensation term module 1754 displays the term with  
default values (Amount 0, Currency from the last term or the default currency for the  
system, interest rate 0, fields for entering a Time Unit Period (see Extensions) with

default Time Unit Type of Month, Term Type, Royalty\_Per\_Sales\_Amount). The enter compensation term module 1754 displays a form for the term given the default term type (see Extensions). It displays a choice of several period types and entry fields that differ depending on the period type the user chooses.

5           Step 5. The Actor enters the standard details for the term and the details for the particular term type (see Extensions), then either repeats from step 3 or ends the use case.

10           Step 6. The enter compensation term module 1754 inserts the new terms into the Licensing Database 12404. The enter compensation term module 1754 extends the use case with the Create Expected Revenue use case, which computes the expected revenue amounts per period for the existing compensation terms.

This use case has a number of extensions:

- ° If the Actor presses F1 or the equivalent, the application extends the use case with the Display Help use case.
- 15           ° If the term type is Ongoing Royalty, the enter compensation term module 1754 displays fields for entering a Recurring Period (see below) and an ongoing royalty type (Per Unit, Per Sales, Per Royalty (Sublicense), default Per Sales.
- 20           ° If the term type is Ongoing Royalty and the Royalty Type is Royalty Per Unit, the enter compensation term module 1754 displays fields for entering a Royalty Unit (default 'Unit'), the Estimated Number of Units per Period (default 0), a table for an escalation schedule (Starting Units, Ending Units, and Amount), and a table for Estimated Units (Period Number, Estimated Units).
- 25           ° If the term type is Ongoing Royalty and the Royalty Type is Royalty Per Sales Amount, the enter compensation term module 1754 displays fields for entering the Percentage, the Net flag (default True), the Estimated Revenue Per Period (default 0), a table for an escalation schedule (Starting Amount, Ending Amount, and Royalty Percentage), and a table for Estimated Revenue (Period Number, Estimated Revenue).

- ° If the term type is Ongoing Royalty and the Royalty Type is Royalty Per Royalty Amount, the enter compensation term module 1754 displays the same fields as for Royalty Per Sales Amount.
- ° If the Term Type is Minimum Guarantee, the enter compensation term module 1754 displays fields for entering a Time Unit Period (see below), a Due Date, and a table for an escalation schedule (Start Date, End Date, Amount).
- ° If the Term Type is Advance, the enter compensation term module 1754 displays the Due Date of the advance and the Refundable flag.
- ° If the Term Type is Fee and the Fee Type is Ongoing Fee, the enter compensation term module 1754 displays fields for entering a Time Unit Period (see below).
- ° If the Term Type is Fee and the Fee Type is Lump Sum, the enter compensation term module 1754 displays the Due Date.
- ° If the Actor must enter a Recurring Period, the enter compensation term module 1754 displays the Time Unit Type, the Start Date, the Recurrence Rate, the Starting Interval, the Interval Unit Type, and the choice of ending the cycle with a specific number of recurrences (Recurring Period) or a specific End Date (End Date Period).

### ***Create Expected Revenue***

- 20           The create expected revenue use case 7702 is diagramed in FIG. 77. This use case extends the Enter Compensation Term use case to build a series of expected future revenue payments. In other words, based on the compensation terms of the license agreement, it is possible for the invention to calculate expected revenue to be generated in the future by the license agreement. Such expected future revenue can
- 25           then be compared to actual revenue.

The extending use case passes in the compensation term object identifier. The create expected revenue module 1760 queries the compensation term and its components and generates the appropriate payments expected from that term.

5        Operation of the create expected revenue module 1760 is further described below.

Step 1. The create expected revenue module 1760 queries the Compensation Term in question from the Licensing Database 1204 to get the Amount and subtype of the term.

10        Step 2. For each period that is part of the structure of the compensation term, the create expected revenue module 1760 creates an Expected License Revenue object in the Licensing Database 1204, giving it a payment number and a due date as well as the expected amount of the payment.

15        Step 3. The create expected revenue module 1760 iterates through the expected revenues calculated in step 2 to subtract advance payment amounts from expected royalties and to reset Minimum Guarantees to the amount needed to bring the guarantee up to the guaranteed amount.

This use case has a number of extensions:

- 20        ° For an Ongoing Royalty, the create expected revenue module 1760 queries the time period, the scaling royalty structure, and the royalty type.
- 25        ° For a Royalty Per Unit, the Create expected revenue module 1760 queries the Estimated Units Per Period and the set of period estimates. Using this information, the time period, and the scaling royalty structure, the Create expected revenue module 1760 computes the series of royalty payments due through the expiration of the license. The Description is "Per-unit royalty for the period from <start> to <end> at <Amount> <currency symbol> per <unit> [using scaled royalty from <starting units> to <ending units>]". The <start> and <end> are the computed period boundaries, the Amount is the term amount, the <currency symbol> is the term Unit Symbol, the <unit> is the Unit from the Ongoing Royalty, and the optional string

exists only for scaling royalties and lists the Starting Units and Ending Units from the Ongoing Royalty.

5           °       For a Royalty Per Sales Amount, the Create expected revenue module 1760 queries the Estimated Revenue Per Period and the set of period estimates. Using this information, the time period, and the scaling royalty structure, the Create expected revenue module 1760 computes the series of royalty payments due through the expiration of the license. The Description is "Revenue-based [sublicense] royalty for the period from <start> to <end> at <Percentage> per <currency unit> of revenue [using scaled royalty from <starting units> to <ending units>]". The optional "sublicense" string appears when the object is a Royalty Per Royalty Amount. The 10       <start> and <end> are the computed period boundaries, the Percentage is the Royalty Per Sales Amount Percentage, and the <currency unit> is the Unit Symbol for the term currency. The optional string exists only for scaling royalties and lists the Starting Units and Ending Units from the Ongoing Royalty.

15           °       For a Minimum Guarantee, the Create expected revenue module 1760 queries the time period, the Due Date, and the scaling guarantee structure. Using this information, the Create expected revenue module 1760 computes the series of guarantee dates, setting the Expected Revenue Amount to the term Amount given the scaling table based on the Due Date. The Description is "Minimum Guarantee for 20       period from <start> to <end> using scaled guarantee for <Due Date>", where <start> and <end> are the guarantee period boundaries and the <Due Date> is the Due Date of the actual scaling term used.

25           °       For an Advance, the Create expected revenue module 1760 queries the due date. The Create expected revenue module 1760 sets the Amount to the term amount and the Due Date to the term Due Date. The Description is "Advance on royalties due to be paid on <Due Date>".

          °       For a Fee, the Create expected revenue module 1760 queries the fee type.

- 5           °       For an Ongoing Fee, the Create expected revenue module 1760 queries the time period. Using this information, the Create expected revenue module 1760 computes the series of fee payments due through the expiration of the license. The Amount is the term amount. The Description is "Ongoing fee for period from <start> to <end>", where <start> and <end> are the time period boundaries.
- °       For a Lump Sum Fee, the Create expected revenue module 1760 sets the Amount to the term amount and the Due Date to the Lump Sum Fee due date. The Description is "Lump-sum fee due on <Due Date>".
- 10          °       For Other Compensation, the Create expected revenue module 1760 queries the Description and Due Date and sets the Expected Revenue Amount to 0 and the Due\_Date to the Other Compensation Due Date. The Description is the term Description.

### *Modify Compensation Term*

15       The modify compensation term use case 7802 is diagramed in FIG. 78. The Modify License Agreement use case uses this one to change existing compensation terms that belong to a license agreement in the Licensing Database 1204. The Licensing Administrator selects a particular term and modifies its attributes. The operation of this use case is further described below.

20       Step 1. The using use case passes in the object identifier for the license agreement in which the License Administrator wants to change compensation terms.

          Step 2. The modify compensation term module 1762 displays a list of the current set of compensation terms for this license in order of term number, displaying the term Description and the Term Type.

          Step 3. The License Administrator selects a term for modification.

25       Step 4. The modify compensation term module 1762 displays the current values. It displays a data entry form for the term given the term type (see Extensions) that display the current values for the selected term.



Step 5. The License Administrator changes any of these values, then ends the use case.

Step 6. The modify compensation term module 1762 updates the term in the Licensing Database 1204.

5           This use case has a number of extensions:

- °           If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.
- °           If the term type is Ongoing Royalty, the Modify compensation term module 1762 displays fields for entering a Time Unit Period (see below) and an ongoing royalty type (Per Unit, Per Sales, Per Royalty (Sublicense), default Per Sales.  
10           If the term type is Ongoing Royalty and the Royalty Type is Royalty Per Unit, the Modify compensation term module 1762 displays fields for entering a Royalty Unit (default 'Unit'), the Estimated Number of Units per Period (default 0), a table for an escalation schedule (Starting Units, Ending Units, and Amount), and a  
15           table for Estimated Units (Period Number, Estimated Units).
  - °           If the term type is Ongoing Royalty and the Royalty Type is Royalty Per Sales Amount, the Modify compensation term module 1762 displays fields for entering the Percentage, the Net flag (default True), the Estimated Revenue Per Period (default 0), a table for an escalation schedule (Starting Amount, Ending Amount, and  
20           Royalty Percentage), and a table for Estimated Revenue (Period Number, Estimated Revenue).
    - °           If the term type is Ongoing Royalty and the Royalty Type is Royalty Per Royalty Amount, the Modify compensation term module 1762 displays the same fields as for Royalty Per Sales Amount.
  - °           If the Term Type is Minimum Guarantee, the Modify compensation term module 1762 displays fields for entering a Time Unit Period (see below), a Due Date, and a table for an escalation schedule (Start Date, End Date, Amount).
  - °           If the Term Type is Advance, the Modify compensation term module 1762 displays the Due Date of the advance and the Refundable flag.

- If the Term Type is Fee and the Fee Type is Ongoing Fee, the Modify compensation term module 1762 displays fields for entering a Time Unit Period (see below).

- If the Term Type is Fee and the Fee Type is Lump Sum, the Modify compensation term module 1762 displays the Due Date.

- If the License Administrator must enter a Time Unit Period, the Modify compensation term module 1762 displays the Time Unit Type, the Start Date, the Recurrence Rate, the Starting Interval, the Interval Unit Type, and the choice of ending the cycle with a specific number of recurrences (Recurring Period) or a specific End Date (End Date Period).

### ***Remove Compensation Term***

The remove compensation term use case 7902 is diagramed in FIG. 79. The Modify License Agreement use case uses this one to remove existing compensation terms that belong to a license agreement in the Licensing Database 1204. The Licensing Administrator selects a particular term and removes it. The operation of this use case is further described below.

Step 1. The using use case passes in the object identifier for the license agreement in which the License Administrator wants to change compensation terms.

Step 2. The Remove compensation term module 1764 displays a list of the current set of compensation terms for this license in order of term number, displaying the term Description and the Term Type.

Step 3. The License Administrator selects a term to remove.

Step 4. The Remove compensation term module 1764 deletes the information from the Licensing Database and moves it to the Recycle Bin (implementation may differ regarding deferring the actual SQL deletes, and there is no notification of commit to the Administrator).

### *Query License*

The query license use case 8002 is diagramed in FIG. 80. According to this use case, the Auditor, the Data Entry Clerk, or the License Administrator (and potentially any User) queries a license from the Licensing Database 1204 based on any of the structured fields of the license. The system displays the details of the license agreement, including a list of asset packages, the parties to the agreement, a list of compensation terms, a list of royalty statements, etc. The operation of this use case is further described below.

Step 1. The Auditor starts the transaction. This can be accomplished in a number of ways, such as by selecting Tools, Find, License Agreement from the menu bar (FIG. 67).

Step 2. The Query license module 1768 displays a query form 8102 (FIG. 81) with any combination of the following structured fields available for query specification: object identifier, Agreement ID, Name, Title, Licensee, Licensor, Description, Effective Date, Exclusivity, Assignable, Transferable, Revocable, Confidential, Terminated, Expiration Date, Perpetual, Infringement Based, Can Sublicense, Annual Audit, Audit at Licensee Expense, Favored Nation, Improvements, Grant Back, Withholding Percentage, Asset Package Name, Field of Use, Territory, Party Name, etc. The example query form 25702 in FIG. 257 shows only a subset of these fields.

Step 3. The Auditor enters the search criteria.

Step 4. The Query license module 1768 queries all the licenses that meet the query criteria from the licensing database 1204 and for which the Auditor has Read access. The Query license module 1768 displays the licenses, preferably showing all of the fields in step 2 except for Party Name, Asset Package Name, Territory, and Field of Use, which are extensions. The Query license module 1768 orders the licenses by Agreement ID.

Step 5. The Auditor ends the transaction.

This use case has a number of extensions:

- 5           ◦       If the license agreement has any asset packages to which the Auditor has Read access or which the Auditor owns, the Query license module 1768 displays the package name, whether the package is a cross-license package, and the Limitations field in a list ordered by name. If the Auditor selects a package, the Query license module 1768 extends the use case with the Modify IP Asset Package use case.
- 10           ◦       If the license agreement/package combination has any territorial restrictions, the Query license module 1768 displays the territory's Abbreviation, Full Name, and Description in a list ordered by Abbreviation. The Auditor cannot change territories here but must instead use the Modify License Agreement use case.
- 15           ◦       If the license agreement/package combination has any field-of-use restrictions, the Query license module 1768 displays the field-of-use Display Name and Description in a list ordered by name. The Auditor cannot change territories here but must instead use the Modify License Agreement use case.
- 20           ◦       If the license agreement has any parties, the Query license module 1768 displays the entity name and role name of the parties in a list ordered by role name and entity name. If the Auditor selects a party, the Query license module 1768 extends this use case with the Modify Entity use case.
- 25           ◦       If the license agreement has any compensation terms, the Query license module 1768 displays the term number, term type, and description in a list ordered by term number. If the Auditor selects a term, the Query license module 1768 extends this use case with the Modify Compensation Term use case.
- If the license agreement has any royalty statements, the Query license module 1768 displays the royalty statement reporting period in a list ordered by period. If the Auditor selects a statement, the Query license module 1768 extends this use case with the Modify Royalty Statement use case.
- If Infringement Based is True, the Query license module 1768 displays the two Infr Based Agreement fields Infringement Release Granted and Infringement Comment.

- If the Auditor selects a queried license agreement row, the Query license module 1768 extends the use case with the Modify License Agreement use case.
- If the Auditor presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Modify License Agreement*

The modify license agreement use case 8202 is diagramed in FIG. 82. According to this use case, the Data Entry Clerk modifies a license agreement in the Licensing database 1204. Modifying the agreement entails entering/revising structured data (name, description, effective date, and so on). The Data Entry Clerk can edit or add new compensation terms that describe the fee and royalty structure of the agreement. The Data Entry Clerk can revise the links between the agreement and the set of asset packages representing the licensed IP assets. The Data Entry Clerk can revise the links between the agreement and the set of entity roles that identify the various parties to the agreement (licensor, licensees, attorneys, and so on). This use case is further described below.

Step 1. The License Administrator begins the transaction.

Step 2. The Modify license agreement module 1758 displays the License Agreements in the Licensing Database.

Step 3. The License Administrator chooses a license agreement to modify.

Step 4. If the License Administrator has write permission on the selected document, the Modify license agreement module 1758 displays the license agreement in a data entry form such as that used in the Enter License Agreement use case and shows the current values of all fields. See FIG. 69.

Step 5. The License Administrator modifies any of the structured fields and linked objects of the license agreement in any order:

Step 5.1. The License Administrator modifies the structured fields of the agreement: the Agreement ID, the Name, Description, Effective Date, Expiration Date, Exclusivity, Assignable, Transferable, Revocable, Confidential, and Terminated.

5 Step 5.2. The License Administrator modifies the structured fields of the license agreement: Perpetual, Infringement Based (see Extensions), Can Sublicense, Annual Audit, Audit at Licensee Expense, Favored Nation, Improvements, Grant Back, and Withholding Percentage.

10 Step 5.3. The License Administrator uses the Link to Asset Packages use case to modify the set of asset packages for the license.

Step 5.4. The License Administrator uses the Enter Compensation Term use case to create the compensation terms for the license. This step can be repeated.

15 Step 5.5. The License Administrator uses the Remove Compensation Term use case to remove any unwanted compensation terms from the license to the Recycle Bin. This step can be repeated.

20 Step 5.6. The License Administrator uses the Link to Party use case to modify the set of parties to the agreement. Required parties include one licensor and at least one licensee. You can also add other parties such as attorneys and other roles entered through the Add Role use case by the System Administrator. This step can be repeated.

Step 5.7. The License Administrator selects a different Security Class for the document from a list of available security classes.

Step 6. The License Administrator commits the transaction.

25 Step 7. The Modify license agreement module 1758 updates the Agreement, License Agreement, and Infr Based Agreement objects in the Licensing Database 1204 and confirms the commit to the License Administrator.

This use case has a number of extensions:

- If the asset package or packages the License Administrator wants to add do not exist, extend the use case with the Create IP Asset Package use case to create the appropriate package(s).

5           ◦ If the correct entity the License Administrator wants to add does not exist, extend the use case with the Enter Entity use case to create the appropriate entity.

- If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### ***Remove License Agreement***

10           The remove license agreement use case 8302 is diagramed in FIG. 83. This is preferably an administrator use case. According to this use case, the license system Administrator removes a license agreement and its related data from the Licensing Database 1204, including infringement information, compensation terms, expected revenue, royalty statements, party links, asset package links, territorial restrictions, and  
15           field-of-use restrictions. The operation of this use case is described below.

            Step 1. The License system Administrator begins the transaction.

            Step 2. The Remove license agreement module 1705 displays a list of all the licenses in the Licensing Database, displaying the document ID, the Agreement ID, and the Name.

20           Step 3. The License system Administrator selects one or more license agreements and removes them.

            Step 4. The Remove license agreement module 1705 uses the Remove Royalty Statement use case to remove the Royalty Statements with the agreements.

25           Step 5. The Remove license agreement module 1705 deletes the information from the Licensing database 1204 and moves it to the Recycle Bin (implementation may differ regarding deferring the actual SQL deletes, and there is no notification of commit to the License system Administrator).

This use case has a number of extensions:

- At any time until clearing the Recycle Bin, the administrator can restore the removed License Agreement and its related information.
- If the License system Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

5

### *Print License*

The print license use case 8402 is diagramed in FIG. 84. This use case extends the Print Object use case. The System prints a License Agreement, including all information in the Licensing Database 1204 about the agreement, its parties, its compensation terms, its asset packages (with territorial and field-of-use restrictions), its royalty statements, and the expected revenue and payments received to date. The operation of this use case is further described below.

10

Step 1. The Actor in the Print Object use case selects a License Agreement and prints it.

15

Step 2. The Print license module 1712 prints a formatted report with the Document ID, the Agreement ID, the Name, the Description, the Effective Date, the Expiration Date, and the various clause flags as check boxes.

Step 3. The Print license module 1712 prints the parties to the agreement, with name, organization, and primary contact information.

20

Step 4. The Print license module 1712 prints the compensation terms of the agreement, printing the Term Number, Term Type, and the Description of each term along with the currency symbol and the Amount and any Due Date for the term.

Step 5. The Print license module 1712 prints the Asset Packages for the license using the Print Asset Package use case.

25

Step 6. The Print license module 1712 prints the Royalty Statements received for the license using the Print Royalty Statement use case.



Step 7. The Print license module 1712 prints the Expected Revenues and the Payments linked to them, printing the Payment Number, Amount, and Due Date for each Revenue and the Payment ID, Receipt Date, Currency Unit Symbol, and Amount Allocated for each Payment.

5

### *Administer Territories*

The administer territories use case 8502 is diagramed in FIG. 85. This is preferably an administrator use case. According to this use case, the system Administrator creates, modifies, and/or removes territories from the Licensing Database 1204. The system comes preinstalled with the territories Worldwide, USA, Japan, and EU. The operation of this use case is further described below.

10

Step 1. The System Administrator starts the transaction.

Step 2. The Administer territories module 1709 displays a list of Territories, displaying the Territory ID, Abbreviation (such as EUSA), Full Name (such as East Coast of the United States of America), and Description (such as "the region including all eastern states of the United States of America").

15

Step 3. The System Administer takes one of the following actions:

Step 3.1. The System Administrator creates a new Territory, entering the Abbreviation, Full Name, and Description (the Administer territories module 1709 generates the Territory ID).

20

Step 3.2. The System Administrator selects one of the current Territories and modifies the Abbreviation, the Full Name, or the Description.

Step 3.3. The System Administrator selects one of the current Territories and sets that territory to be the default.

25

Step 3.4. The System Administrator selects one of the current Territories and removes it.

Step 4. The System Administrator ends the transaction.

Step 5. The Administer territories module 1709 commits the changes in the Licensing Database 1204. The Administer territories module 1709 confirms the commit to the System Administrator.

This use case has a number of extensions:

- 5           °       If the System Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.
- °       If removing a territory, the Administer territories module 1709 deletes the information from the Licensing database 1204 and moves it to the Recycle Bin (implementation may differ regarding deferring the actual SQL deletes, and there is  
10       no notification of commit to the System Administrator). The Administer territories module 1709 confirms the commit to the System Administrator. At any time until clearing the Recycle Bin, the administrator can restore the removed Territory and its related information.

### *Administer Fields of Use*

15           The administer fields of use case 8602 is diagramed in FIG. 86. This is preferably an administrator use case. According to this use case, the system Administrator creates, modifies, and/or removes fields of use from the Licensing Database 1204. The operation of this use case is further described below.

Step 1. The System Administrator starts the transaction.

20           Step 2. The Administer fields of use module 1711 displays a list of Fields of Use, displaying the Field of Use ID, Display Name (for example, "semi"), and Description (for example, "semiconductor applications").

Step 3. The System Administer takes any of the following actions:

25           Step 3.1.       The System Administrator creates a new Field of Use, entering the Display Name and Description (the Administer fields of use module 1711 generates the Field of Use ID).

Step 3.2. The System Administrator selects one of the current Fields of Use and modifies the Display Name or the Description.

Step 3.3. The System Administrator selects one of the current Fields of Use and sets it to be the default.

5 Step 3.4. The System Administrator selects one of the current Fields of Use and removes it.

Step 4. The System Administrator ends the transaction.

Step 5. The Administer fields of use module 1711 commits the changes in the Licensing Database 1204. The Administer fields of use module 1711 confirms the  
10 commit to the System Administrator.

This use case has a number of extensions:

- ° If the System Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.
- ° If removing a Field of Use, the Administer fields of use module 1711  
15 deletes the information from the Licensing database 1204 and moves it to the Recycle Bin (implementation may differ regarding details for deferring the actual SQL deletes, and there is no notification of commit to the System Administrator). The Administer fields of use module 1711 confirms the commit to the System Administrator. At any time until clearing the Recycle Bin, the administrator can restore the removed Field  
20 of Use and its related information.

### *Display License Agreements*

The display license agreements use case 15302 is diagramed in FIG. 153. This use case displays a table of license agreements with the basic data for each agreement. The Actor may select one or more agreements and conduct further  
25 operations such as Modify License Agreement. The operation of this use case is further described below.

Step 1. The Actor starts the transaction.

Step 2. The display license agreements module 15304 displays a table of all the License Agreements to which the Actor has read access. Each row of the table corresponds to a License Agreement. The columns include the License Agreement ID, the Name, the Description, the Effective Date, the Expiration Date, the Withholding Percentage, and a series of checkbox items: Exclusivity, Assignable, Transferable, Revocable, Confidential, Terminated, Perpetual, Infringement Based, Can Sublicense, Annual Audit, Audit At Licensee Expense, Favored Nation, Improvements, and Grant Back.

Step 3. The Actor ends the transaction.

This use case has a number of extensions:

- ° If the Actor wants to enter a new license, the Enter License Agreement use case extends this one in a separate transaction. When the extending use case ends, the display license agreements module 15304 displays the new license and its fields in the table.

- ° If the Actor wants to see or modify the details of the license agreement, the Modify License Agreement use case extends this one in a separate transaction. The Actor selects a license agreement from the table and initiates the extending use case. The saved changes from the extending use case appear in the table when that use case ends.

- ° If the Actor wants to print a license agreement, the Print License use case extends this one. The Actor selects one or more license agreements from the table and initiates the extending use case.

- ° If the Actor is the License Administrator and wants to remove the selected licenses, the Remove License Agreement use case extends this one. The Actor selects one or more license agreement from the table and initiates the extending use case. When the extending use case returns, the display license agreements module 15304 removes the selected license or licenses from the table.

*Enter Adjustment*

The enter adjustment use case 15702 is diagramed in FIG. 157. The Enter License Agreement and Modify License Agreement use cases use this one to enter adjustments to compensation terms in a license agreement in the Licensing Database 1204. The Licensing Administrator or Data Entry Clerk creates an adjustment with the details appropriate to the particular kind of adjustment. Adjustments have a description, an amount, a currency, and a due date. A minimum guarantee also has a time period and possible escalations. An advance may be refundable or not.

The adjustments modify the revenue stream that compensation terms generate, and the Link Adjustment to Terms use case relates an adjustment to the terms that generate the revenue that it adjusts.

It will be illustrative to consider selected background items. A license agreement structures its revenue stream based on a combination of fees and royalty payments (and possibly some other kinds of compensation). The agreement also contains terms that affect the revenue from these terms, the minimum guarantee and the advance.

A minimum guarantee is an amount of money that the licensor must receive by the end of the guarantee period, usually one year but sometimes quarterly or some other period. If the revenue from certain compensation terms does not add up to this amount, the licensee must make up the difference.

An advance is an amount paid to the licensor by the licensee in advance of revenue (usually royalties). The licensee then subtracts the advance amount from the compensation due until the entire amount of the advance is accounted for. From that point on, the licensee then begins payments to the licensor.

Any given adjustment may cover any number of different compensation terms for the same license. For example, an advance may cover an initial fee and a series of royalty payments. A minimum guarantee could cover two or three different royalty terms but not a maintenance fee.

The operation of this use case is further described below.

Step 1. The using use case passes in the object identifier for the license agreement to which the Data Entry Clerk or License Administrator (the Actor) wants to add adjustments.

5 Step 2. The enter adjustment module 15702 displays a list of the current set of adjustments for this license in preferably order of term number, displaying the Description, the Amount with its currency symbol, the Due Date, and the Adjustment Type.

Step 3. The Actor creates a new adjustment and selects one of the term types (Minimum Guarantee or Advance).

10 Step 4. The enter adjustment module 15702 displays the adjustment with default values (Amount 0, Currency from the last term or the default currency for the system, and Due\_Date empty). It displays a form for the term given the default term type (see Extensions).

15 Step 5. The Actor enters the standard details for the adjustment and the details for the particular term type (see Extensions), then either repeats from step 3 or ends the use case.

Step 6. The enter adjustment module 15702 inserts the new terms into the Licensing Database 1204.

This use case has a number of extensions.

20 ° If the Actor presses F1 or the equivalent, the application extends the use case with the Display Help use case.

° If the Term Type is Minimum Guarantee, the Actor uses the Enter Recurring Time Period use case to enter a Recurring Period (see below) or enters a Single Date. Optionally, the Actor enters a table for an escalation schedule (an Open Ended Period and an Amount for each row of the table). The enter adjustment module 25 15702 inserts the items into the Minimum\_Guarantee table and the Date\_Amount\_Interval table.

° If the Term Type is Advance, the enter adjustment module 15702 displays a field for entering a Single Date representing the due date of the advance and

a field for the Refundable flag. The enter adjustment module 15702 inserts these into the Advance table.

° If the Actor must enter a Recurring Period for a minimum guarantee, the enter adjustment module 15702 displays the Time Unit, the Start Date, the Recurrence Rate, the Starting Interval, the Interval Unit, and the choice of ending the cycle with a specific number of recurrences (Count Limited Recurring Period) or a specific End Date (Date Limited Recurring Period).

### *Modify Adjustment*

The modify adjustment use case 16102 is diagramed in FIG. 161. The Modify License Agreement use case uses this one to change existing adjustments that belong to a license agreement in the Licensing Database 1204. The Licensing Administrator selects a particular adjustment and modifies its attributes.

The Licensing Administrator also modifies the links between adjustments and compensation terms for the license agreement in the Link Adjustment to Terms use case. See the Enter Adjustment use case for background material on adjustments.

The operation of this use case is further described below.

Step 1. The using use case passes in the object identifier for the license agreement in which the License Administrator wants to change adjustments.

Step 2. The modify adjustment module 16104 displays a list of the current set of adjustments for this license in order of term number, displaying the term Description and the Term Type.

Step 3. The License Administrator selects an adjustment for modification.

Step 4. The modify adjustment module 16104 displays the current values for the selected adjustment through a data entry form for the adjustment, including due date, currency, amount, or description.

Step 5. The License Administrator changes any of these values, then ends the use case.

Step 6. The modify adjustment module 16104 updates the adjustment in the Licensing Database 1204.

- If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

5           ◦ If the Term Type is Minimum Guarantee, the modify adjustment module 16104 displays fields for entering a Recurring Period (see below) and a table for an escalation schedule (an Open Ended Period and an Amount). The License Administrator may add or remove intervals from the escalation schedule or may change the amount or start date for an existing interval.

10           ◦ If the Term Type is Advance, the modify adjustment module 16104 displays the Due Date of the advance and the Refundable flag.

          ◦ If the Actor must enter a Recurring Period, the modify adjustment module 16104 displays the Time Unit, the Start Date, the Recurrence Rate, the Starting Interval, the Interval Unit, and the choice of ending the cycle with a specific number of recurrences (Count Limited Recurring Period) or a specific End Date (Date Limited Recurring Period).

15           ◦ If the modification to the adjustment affects the expected revenue for the license agreement, then all the expected license revenues not linked to any payment and affected by the adjustment are removed and replaced with the changed values.

20

### ***Link to Adjustment***

The link to adjustment use case 16202 is diagramed in FIG. 162. According to this use case, a License Administrator selects one or more adjustments from a list. The link to adjustment module 16202 creates a link between the input compensation

25           term and the selected adjustments.



It will be illustrative to consider selected background items. The invention allows one to adjust the revenue stream from a compensation term through advances or minimum guarantees. These are adjustment objects that are part of the license.

5       The advance is a payment from the licensee to the licensor in anticipation of revenue from the license. When the revenue comes due, the amount from the compensation terms covered by the advance is reduced by the amount of the advance until the revenue accounts for the entire advance.

10       The minimum guarantee is a floor for revenue during a specific period, usually a year. If the revenue for that period does not come up to the guaranteed amount, the difference becomes due as a separate payment.

Advances and minimum guarantees may apply to only certain compensation terms associated with the license. Linking the compensation terms to the adjustments lets you specify exactly how to adjust the revenue generated from the compensation terms in the Create Expected Revenue use case.

15       The operation of this use case is further described below.

Step 1. The License Administrator starts the use case with an open transaction containing a compensation term that will link to the adjustments. The extended use case supplies the object identifier for the agreement.

20       Step 2. The link to adjustment module 16204 displays the types and descriptions of the adjustments for the license that owns the compensation term in a list ordered by adjustment number within the license, displaying any existing links.

Step 3. The License Administrator selects one or more adjustments from the list.

25       Step 4. The adjustment module 16204 creates and displays a link for each adjustment the License Administrator selects.

Step 5. The Licensing Database 1204 links the adjustments to the compensation term through the Term\_Adjustment association table.

This use case has a number of extensions.

- The License Administrator can cancel the use case at any time. The result will be that the compensation term is unchanged from its status before using the use case.

- If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

- If the License Administrator selects an existing link from a compensation adjustment to a compensation term, he or she can delete the link.

### *Remove Compensation Adjustment*

The remove compensation adjustment use case 16302 is diagramed in FIG. 163. The Modify License Agreement use case uses this one to remove existing compensation adjustments that belong to a license agreement in the Licensing Database 1204. The operation of this use case is further described below.

Step 1. The using use case passes in the object identifier for the license agreement in which the License Administrator wants to change compensation adjustments.

Step 2. The remove compensation adjustment module 16304 displays a list of the current set of compensation adjustments for this license in order of adjustment number, displaying the term Description and the Adjustment Type.

Step 3. The License Administrator selects an adjustment to remove. In an embodiment, the adjustment cannot link to any compensation terms of the license.

Step 4. The remove compensation adjustment module 16304 deletes the information from the Licensing Database 1204 and moves it to the Recycle Bin.

Preferably, if the selected compensation adjustment has any compensation term objects associated with it, the remove compensation adjustment module 16304 refuses to remove the compensation adjustment and instead displays an error message, "Cannot remove this compensation adjustment because there are links to compensation terms of this license."

### *Royalty Statements Use Cases*

5 A royalty statement is a document that a licensee submits to a licensor to specify the licensed royalties owed for a given royalty period. A royalty statement includes a number of royalty statement details. Each royalty statement detail lists a product, a number of units sold, an amount of revenue, and a calculated royalty amount.

10 The Royalty Statements View 8702 contains two panes, a license agreement spreadsheet 8704 and a royalty statement spreadsheet 8706. FIG. 87A. The agreement pane 8704 lets you select a license, and the royalty statement pane 8706 displays the royalty statements for that license. You add and modify statements by double-clicking on a blank row or an existing row, as with all the other views.

### *Enter Royalty Statement*

15 The enter royalty statement use case 8701 is diagramed in FIG. 87B. According to this use case, the Data Entry Clerk queries a license, then enters information about a royalty statement that applies to that license into the database, then enters a series of detail items, one for each product detail on the royalty statement. The new document has the classifier the Data Entry Clerk sets for it. The operation of this use case is further described below.

20 Step 1. The Data Entry Clerk begins the transaction to enter a new royalty statement. The royalty statement view 8702 (FIG. 87A) is displayed.

Step 2. The Data Entry Clerk uses the Query License use case to identify and select the license to which the new royalty statement applies. Alternatively, the Data Entry Clerk can select a license in the license agreement pane 8704 of the royalty statement view 8702.

25 Step 3. The Enter royalty statement module 1770 creates an object identifier for the new royalty statement that it uses to relate the statement to the queried license

agreement and to the statement details. It then links the royalty statement to the queried license.

Step 4. A royalty statement dialog 8802 (FIG. 88) is displayed. The royalty statement dialog 8802 has a general tab 8804 and a details tab 8806. In the general  
5 tab 8804, the Data Entry Clerk enters the Fixed Time Interval for the reporting/statement period (StartDate and End Date). The Data Entry Clerk enters the royalty statement due date, receipt date, and (optionally) an investigation window date and a licensor invoice number. The Data Entry Clerk optionally sets a security  
10 classification from a list of available Write classifications from the IPAM Security Subsystem 1602, to indicate the persons who have access to this royalty statement record.

Step 5. In the details tab 8806 (see FIG. 89), for each detail item on the royalty statement, the Data Entry Clerk creates a new detail for the royalty statement record by pressing the Add Detail button 8904. This results in displaying a detail  
15 dialog 9002. The Data Entry Clerk enters the product (selecting from a list of products associated with the licensee), the number of units sold of the product, the revenue received from the sale and the currency unit for the revenue, and the royalty due for the product, where this information is obtained from the royalty statement provided by the licensee. All details for the royalty statement are listed in a details  
20 area 8902 (FIG. 89).

Step 6. The Data Entry Clerk commits the transaction.

Step 7. The Enter royalty statement module 1770 inserts the Royalty Statement and its details into the licensing database 1204. The Enter royalty statement module 1770 requests the IPAM Security System 1602 to set the document  
25 classification. The enter royalty statement module 1770 confirms the commit to the Data Entry Clerk.

At any time, if the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Context-Sensitive Help use case.

### *Modify Royalty Statement*

The modify royalty statement use case 9102 is diagramed in FIG. 91. According to this use case, the License Administrator queries a license, then modifies any of the attributes of any of the linked royalty statements, and/or modifies any of the statement details, one for each product detail on the statement. The operation of this use case is further described below.

Step 1. The License Administrator begins the transaction.

Step 2. The Modify royalty statement module 1766 displays the licenses to which the License Administrator has read access, displaying the object identifier, the Agreement ID, the Name, and the Description, in a form similar to that shown in FIG. 87A.

Step 3. The License Administrator selects a license from the license pane 8704.

Step 4. The Modify royalty statement module 1766 displays a list of existing royalty statements attached to the license in the royalty statement pane 8706. The Modify royalty statement module 1766 displays only those royalty statements to which the License Administrator has Read access.

Step 5. The License Administrator selects a royalty statement for modification from the royalty statement pane 8706.

Step 6. If the License Administrator has Write access to the royalty statement, the Modify royalty statement module 1766 displays the current field values in a form similar to that shown in FIG. 88.

Step 7. The License Administrator may modify the royalty statement Invoice Number, Due Date, Receipt Date, or the Investigation Window Date in any order.

Step 8. If the details tab 8806 is selected, the Modify royalty statement module 1766 displays the details for the selected royalty statement, displaying the product, the Units Sold, the Currency, and the Revenue.

Step 9. For each detail item on the royalty statement, the License Administrator can modify the name of the product, the number of units sold, or the revenue received and its currency unit, in any order.

Step 10. The License Administrator commits the transaction.

5 Step 11. The Modify royalty statement module 1766 updates the royalty statement.

Step 12. If the License Administrator updated the time interval for the reporting period, the Modify royalty statement module 1766 creates a new time interval and adds the new period to the Licensing Database 1204 and removes the previous one if it is not linked to any other statements, making the relevant changes to the Time Period extent as well.

10 Step 13. For each detail modified, the Modify royalty statement module 1766 updates the detail.

Step 14. The Modify royalty statement module 1766 confirms the commit to the License Administrator.

15 This use case has a number of extensions:

° If the License Administrator wants to move the Royalty Statement to a different license, he or she drags the Royalty Statement to the license and drops it. The Modify royalty statement module 1766 changes the License ID in the Royalty Statement.

20 ° If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Context-Sensitive Help use case.

### *Query Statement*

25 The query statement use case 9202 is diagramed in FIG. 92. According to this use case, the Auditor enters search criteria for the statement, and the Query statement module 1772 displays the statement from the Licensing Database 1204. The Auditor can select a statement to see the statement details, and he or she can select a detail to

-105-

see any payment allocations for the detail. The operation of this use case is further described below.

Step 1. The Auditor starts the transaction.

5       Step 2. The Query statement module 1772 displays the licenses to which the Auditor has read access, displaying the object identifier, the Agreement ID, the Name, and the Description in a format similar to that shown in FIG. 87A.

Step 3. The Auditor selects a license for query of statements.

10       Step 4. The Query statement module 1772 displays a query form that lets the Auditor enter selection conditions on the set of statements, including object identifier, the reporting period Start and End Dates, the Invoice Number, the Due Date, the Receipt Date, and/or the Investigation Window Date.

Step 5. The Auditor enters query criteria and starts the query.

15       Step 6. The Query statement module 1772 displays the royalty statements for the license in preferably reverse chronological order to which the Auditor has Read access, displaying the object identifier, the reporting period Start and End Dates, the Invoice Number, the Due Date, the Receipt Date, and the Investigation Window Date.

Step 7. The Auditor selects a statement.

20       Step 8. The Query statement module 1772 queries all the details of the selected royalty statement from the Licensing Database 1204 and displays them, if any, showing the Product, the Units Sold, the Revenue and its currency unit symbol, and the Royalty Due, in a form similar to that shown in FIG. 89.

Step 9. The Auditor selects a detail.

25       Step 10. The Query statement module 1772 queries all the payment detail allocations from the Licensing Database 1204 and displays the Amount Allocated for each one with a label that allows the Auditor to see it, if any. The Query statement module 1772 displays the Amount Allocated in the currency of the detail, converting it if the currency of the payment is different from the detail currency unit using the Convert Currency use case. The Query statement module 1772 lists the allocations in preferably Receipt Date order.

Step 11. The Auditor ends the transaction.

### ***Remove Royalty Statement***

The remove royalty statement use case 9302 is diagramed in FIG. 93. Preferably, this is an administrator use case. According to this use case, the system Administrator removes a linked royalty statement, which removes the statement details as well, all from the Licensing Database 1204. This operation is available only through the System Administrator interface to limit the potential for accidental removal of the data. The operation of this use case is further described below.

Step 1. The System Administrator begins the transaction.

Step 2. The Remove royalty statement module 1707 displays a list of existing royalty statements.

Step 3. The System Administrator selects one or more royalty statements and removes them.

Step 4. The Remove royalty statement module 1707 deletes the information from the licensing database 1204 moves it to the Recycle Bin (implementation may differ regarding details on deferring the actual SQL deletes, and there is no notification of commit to the System Administrator).

This use case has a number of extensions:

- If a using use case supplies a license object identifier, the use case shows only the royalty statements for that license (where License\_Agreement\_ID = :License\_ID).

- At any time until clearing the Recycle Bin, the administrator can restore the removed royalty statement.

- If the System Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### ***Print Statement***



The print statement use case 9402 is diagramed in FIG. 94. This use case extends the Print Object use case, and the Print License use case uses it. The Query statement module 1772 prints a Royalty Statement, including all information in the Licensing Database 1204 about the statement, its details, and the allocation of payments to those details. This use case is further described below.

Step 1. The Actor in the Print Object use case or the Print License Use case selects a Royalty Statement and prints it.

Step 2. The Print statement module 1714 prints a formatted report with the Start Date, End Date, Agreement Name, Agreement ID, any Invoice Number, the Due Date, the Receipt Date, and/or the Investigation Window Date.

Step 3. The Print statement module 1714 prints the Royalty Statement Details for the statement. Each detail contains the Product Name, Product Description, Units Sold, Revenue, Currency Unit Symbol, and/or Royalty Due.

Step 4. For each detail, the Print statement module 1714 prints any payment allocations, printing the Payment ID, the Receipt Date, any Invoice Number, the Payor Name, the Amount Allocated, the Currency Unit Symbol, the Payment Amount, and/or the Payment Withheld Amount, preferably ordering the list by receipt date of the payment.

### *Display Royalty Statements*

The display royalty statements use case 15802 is diagramed in FIG. 158. This use case displays a table of licenses and a table of royalty statements with the basic data for each agreement and statement within the agreement. The Actor may select an agreement. The display royalty statements module 15804 then displays all the royalty statements to which the Actor has read access that belong to the selected license agreement. The Actor may then select one or more statements within the agreement.

The operation of this use case is further described below.

-108-

Step 1. The Actor starts the transaction.

Step 2. The display royalty statements module 15804 displays a table of all the License Agreements to which the Actor has read access using the Display License Agreements use case.

5 Step 3. The Actor selects a single license agreement from that table.

Step 4. The display royalty statements module 15804 displays a table of all the Royalty Statements from the Licensing Database 1204 to which the Actor has read access. Each row of the table corresponds to a single Royalty Statement. The columns include the Reporting Period's Start Date and End Date, the Invoice Number, the Due Date, the Receipt Date, and the Investigation Window Date.

10

Step 5. The Actor ends the transaction.

° If the Actor wants to enter a new royalty statement, the Enter Royalty Statement use case extends this one in a separate transaction. When the extending use case ends, the System displays the new statement and its fields in the table.

15 ° If the Actor wants to see or modify the details of the Royalty Statement, the Modify Royalty Statement use case extends this one in a separate transaction. The Actor selects a statement from the table and initiates the extending use case. The saved changes from the extending use case appear in the statement table when that use case ends.

20 ° If the Actor wants to print a royalty statement, the Print Statement use case extends this one. The Actor selects one or more royalty statements from the table and initiates the extending use case.

° If the Actor is the License Administrator and wants to remove the selected statements, the Remove Royalty Statement use case extends this one. The Actor selects one or more royalty statements from the table and initiates the extending use case. When the extending use case returns, the display royalty statements module 15804 removes the selected license or licenses from the table.

25

### *Query Statement*

The query statement use case 15502 is diagramed in FIG. 155. According to this use case, the Auditor enters search criteria for the statement, and the query statement module 15504 displays the statement from the Licensing Database 1204. The Auditor can select a statement to see the statement details, and he or she can select a detail to see any payment allocations for the detail. The operation of this use case is further described below.

Step 1. The Auditor starts the transaction.

Step 2. The query statement module 15504 displays the licenses using the Display Licenses use case.

Step 3. The Auditor selects a license for query.

Step 4. The query statement module 15504 displays a form that lets the Auditor enter any combination of the reporting period Start and End Dates, the Invoice Number, the Due Date, the Receipt Date, and the Investigation Window Date.

Step 5. The Auditor enters values into the query form fields and starts the query.

Step 6. The query statement module 15504 combines the entries into a complete query expression, executes the query in the Licensing Database 1204, then displays the resulting statements in the system that satisfy the query conditions to which the Auditor has read access, ordered by start date.

Step 7. The Auditor selects a statement.

Step 8. The query statement module 15504 queries all the details of the selected royalty statement from the Licensing Database 1204 and displays them in order of Line Number, if any, showing the Product, the Units Sold, the Revenue and its currency unit symbol, and the Royalty Due.

Step 9. The Auditor selects a detail.

Step 10. The query statement module 15504 queries all the payment detail allocations from the Licensing Database 1204 and displays the Amount Allocated for each one with a label that allows the Auditor to see it, if any. The query statement module 15504 displays the Amount Allocated in the currency of the detail, converting

it if the currency of the payment is different from the detail currency unit using the Convert Currency use case. The query statement module 15504 lists the allocations in preferably Receipt Date order.

Step 11. The Auditor ends the transaction.

5

### *Payment Use Cases*

A payment is an amount of money the licensee sends to the licensor based on the amount they owe for fees, royalties, advances, minimum guarantees, and other compensation terms. To produce various reports, the License Administrator must link these payments to expected revenue periods and to royalty statement details.

10

The Payments View 9502 contains two panes, a license agreement spreadsheet 9504 and a payment spreadsheet 9506. The agreement pane 9504 lets you select a license, and the payment pane 9506 displays the payments for that license. You add and modify payments by double-clicking on a blank row or as existing row, as with all the other views.

15

### *Enter Payment*

The payment use case 10002 is diagramed in FIG. 100. According to this use case, the Data Entry Clerk records the details of a payment related to a license in the Licensing Database 1204. The operation of this use case is further described below.

Step 1. The Data Entry Clerk starts the transaction to enter a new payment.

20

Step 2. The Enter payment module 1774 displays a list of licenses in the license pane 9504 of the payments view 9502.

Step 3. The Data Entry Clerk selects a license in the license pane 9504 of the payments view 9502.

25

Step 4. The Enter payment module 1774 displays a list of payments for the license in the payments pane 9506.

Step 5. The Data Entry Clerk signals the desire to add a new payment for the selected license using any procedure described herein.

5 Step 6. The Enter payment module 1774 displays a form 9602 (FIG. 96) for entering the details of a payment and creates a new object identifier for the payment object and a link from that payment to the selected license. The form 9602 includes a general tab 9604, a royalty statement details tab 9606, and an expected revenue tab 9608.

10 Step 7. In the general tab 9604, the Data Entry Clerk enters the Receipt Date, the Amount received, and the Currency in any order. Optionally, the Data Entry Clerk enters a withheld amount (the part of the total amount of the payment that represents a foreign company's withholding of tax due on the payment), an invoice number (for an invoice needed to release the payment from a foreign tax authority), and any late payment interest amount (part of the total amount of the payment that is a penalty for late payment based on the interest rate in the license agreement), again in any order.

15 The Data Entry Clerk selects a Security Class for the payment. The Data Entry Clerk commits the transaction. Data is not typically entered into the royalty statement details tab 9606 and the expected revenue tab 9608, although these tabs are active and accessible according to an embodiment of the invention.

20 Step 8. The Enter payment module 1774 creates a persistent payment and a link to the indicated license. The Enter payment module 1774 confirms the committed transaction to the Data Entry Clerk.

At any time, if the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Modify Payment*

25 The modify payment use case 10102 is diagramed in FIG. 101. According to this use case, the License Administrator modifies the details of a payment related to a license. The License Administrator then optionally links the licensee's payment

to the expected revenue estimates of the license, to the details of a royalty statement, and to the general ledger debit and credit entries. The operation of this use case is further described below.

Step 1. The License Administrator starts the transaction.

5           Step 2. The Modify payment module 1776 displays using the payments view 9502 the licenses in the modify payment module 1776 to which the License Administrator has read access, displaying the object identifier, the Agreement ID, the Name, and the Description.

10           Step 3. The License Administrator selects a license listed in the license agreement pane 9504.

15           Step 4. The Modify payment module 1776 displays in the payments pane 9506 the payments for the license to which the License Administrator has Read access, displaying the object identifier, the Payor Name, the Security Class, the Currency Symbol, The Amount, the Withheld Amount, the Late Payment Interest Amount, the Receipt Date, and/or the Invoice Number.

          Step 5. The License Administrator selects a payment from the payments pane 9506.

20           Step 6. If the License Administrator can write to the payment, the Modify payment module 1776 displays a form 9602 (FIG. 96) for modifying the details of a payment, showing the fields from step 4 with their current values for the selected payment.

          Step 7. The License Administrator optionally modifies any of the fields except for the object identifier and commits the transaction.

25           Step 8. The Modify payment module 1776 updates the payment in the Licensing Database 1204. The Modify payment module 1776 confirms the committed transaction to the Licensing Administrator.

          This use case has a number of extensions:

- °       If the business is linking payments to expected license revenue estimates, the License Administrator uses the Link to Expected Revenue use case to

link the payment to the estimates. The license and payment object identifiers for the selected license and payment pass into the used use case. The used use case creates or removes any links in the Licensing Database.

5           °       If the business is linking payments to royalty statement details, the License Administrator uses the Link to Detail use case to link the payment to the details of one or more royalty statement details. The license and payment object identifiers for the selected license and payment pass into the used use case. The used use case creates or removes any links in the Licensing Database.

10           °       If the business is recording GL entries, the License Administrator enters two General Ledger Entry items, supplying a GL Account Number from a list of valid numbers, an Amount, and an Entry Date (the date on which the GL recorded the payment) for each entry. The License Administrator chooses whether each entry is a debit or a credit entry. (These entries should usually be supplied by the accounting department, which then sends the records on to the licensing department for allocation purposes. The GL feature is optional and configurable.) The Modify payment module  
15       1776 creates a Debit General Ledger Entry or a Credit General Ledger Entry and a General Ledger Entry linked to the payment depending on the Entry Type.

          °       If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

20

### ***Link to Expected Revenue***

          The link to expected revenue use case 10202 is diagramed in FIG. 102. According to this use case, a License Administrator allocates a payment to the expected revenue estimates deriving from the license compensation terms. The link to expected revenue use case 10202 creates a series of links to these estimates in the  
25       Licensing Database 1204. Via these links, it is possible to compare estimated license revenue to actual license revenue/payments. The operation of this use case is further described below.

Step 1. The License Administrator initiates the use case from the calling use case, passing in an object identifier for a payment and an object identifier for a license agreement to which the payment applies.

5 Step 2. The Link to expected revenue module 1778 displays in an estimate pane 9808 of the expected revenue tab 9608 (FIGS. 98 and 99) the Expected Amount, Due Date, and optionally a Description for each expected revenue estimate for the license. Note that there are estimates for each compensation term, and the compensation terms for the license agreement are listed in a term pane 9810.

10 Step 3. The License Administrator selects an estimate and enters an allocation amount for it. This can be done by entering an amount in field 9802, or a percentage of the total payment in field 9804. The total payment amount is displayed in field 9806. This step repeats, as necessary.

15 Step 4. The Link to expected revenue module 1778 stores an allocation amount into the Licensing Database 1204 for each non-zero allocation entered in step 3.

This use case has a number of extensions:

- 20     ° If the License Administrator enters an allocation that violates the constraint on the set of allocations with respect to the total payment, the Link to expected revenue module 1778 displays the error message "Sum of allocations must be less than or equal to total payment" and clears the entered allocation.
- ° If the License Administrator signals completion of linking and there are any nonpositive allocation amounts, the Link to expected revenue module 1778 displays the error message "You must enter a positive allocation for each revenue estimate you select" and puts the focus on the first nonpositive amount in the list.
- 25     ° The currency unit of the payment should match the currency unit of the expected revenue estimate. If not, the Link to expected revenue module 1778 displays a warning message ("Converting currencies from %s to %s") that the amounts are in different currencies and offers the user the chance to use the Convert



-115-

Currency use case to convert one currency to the other. The link to expected revenue module 1778 displays both the amounts in each allocation link.

° If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

5

### *Link to Detail*

The link to detail use case 10302 is diagramed in FIG. 103. This use case extends the Modify Payment use case. While modifying a payment, the License Administrator selects one or more details from a royalty statement. The use case links the selected details to the payment, allocating part of the payment to each of the details. The operation of this use case is further described below.

10

Step 1. The License Administrator initiates the use case from the calling use case, passing in an object identifier for a payment and an object identifier for a license agreement to which the payment applies.

15

Step 2. The Link to detail module 1780 displays the royalty statements for the license in the royalty statement details tab 9606 (FIG. 96). In the example of FIG. 97, there are two royalty statements 9710 and 9712 displayed. Royalty statement 9710 has two details 9714, and royalty statement 9712 has two details 9716. In an embodiment, more complete information is displayed for each detail at this point. Such information may include the line number, product name, units sold, revenue amount, royalty due with currency symbol, etc.

20

Step 3. The License Administrator selects a detail and enters an allocation amount. This can be done by entering an amount (such as in field 9704), or a percentage of the payment total (such as in field 9708). This step repeats as necessary.

25

Step 4. The License Administrator signals completion.

Step 5. The Link to detail module 1780 creates links between the details and the payment.

This use case includes a number of extensions:

- 5           ◦       If the License Administrator enters an allocation that violates the constraint on the set of allocations with respect to the total payment, the Link to detail module 1780 displays the error message "Sum of allocations must be less than or equal to total payment" and clears the entered allocation.
- If the License Administrator signals completion of linking and there are any nonpositive allocation amounts, the Link to detail module 1780 displays the error message "You must enter a positive allocation for each detail you select" and puts the focus on the first nonpositive amount in the list.
- 10          ◦       The currency unit of the payment should match the currency unit of the statement detail. If not, the Link to detail module 1780 displays a warning message ("Converting currencies from %s to %s") that the amounts are in different currencies and offers the user the chance to use the Convert Currency use case to convert one currency to the other. The link to detail module 1780 displays both the  
15           amounts in each allocation link.
- If the License Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### ***Print Payment***

20           The print payment use case 10402 is diagramed in FIG. 104. This use case extends the Print Object use case. The print payment module 1716 prints a Payment, including all information in the Licensing Database 1204 about the payment, its General Ledger entries, and its allocation to royalty statement details and to expected revenue. The operation of this use case is further described below.

25           Step 1. The Actor in the Print Object use case selects a Payment and issues a command to print it.

Step 2. The Print payment module 1716 prints a formatted report with the Payment ID, the Currency Unit Symbol, the Amount, the Withheld Amount, the Late Payment Interest Amount, the Receipt Date, the Invoice Number, etc.

5 Step 3. The Print payment module 1716 prints the General Ledger Entries, printing the GL Account Number, the Account Description, the Amount, the Entry Date, and the Entry Type (credit or debit).

Step 4. The Print payment module 1716 prints the allocations of the payment to royalty statement details, including the Royalty Statement Start and End Dates, the Detail Line Number, the Product Name, the Detail Currency Unit Symbol, Detail  
10 Royalty Due, and the Amount Allocated.

Step 5. The Print payment module 1716 prints the allocations of the payment to expected revenue, including the License Agreement ID and Name, the Term Number, the Expected Currency Unit, the Expected Amount, the Expected Due Date, and the Amount Allocated.

### 15 *Remove Payment*

The remove payment use case 10502 is diagramed in FIG. 105. This is preferably an administrator use case. According to this use case, the System Administrator removes payments and related GL Entries and license and detail allocations from the Licensing Database 1204. This use case is further described  
20 below.

Step 1. The System Administrator begins the transaction.

Step 2. The Remove payment module 1713 displays a list of all the payments in the Licensing Database 1204, displaying the Payment ID, the Payor Name, the Receipt Date, and the Amount.

25 Step 3. The System Administrator selects one or more payments and removes them (that is, issues command(s) to remove them).

Step 4. The Remove payment module 1713 deletes the information from the Licensing database 1204 and moves it to the Recycle Bin (implementation may differ regarding deferring the actual SQL deletes, and there is no notification of commit to the System Administrator).

5 This use case has a number of extensions:

- At any time until clearing the Recycle Bin, the administrator can restore the removed Asset.
- If the System Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

10 *Query Payment*

The query payment use case 10602 is diagramed in FIG. 106. According to this use case, the Auditor queries payments based on payer, on licensees or licensors, on links to royalty statements or expected revenue, on General Ledger accounts, or on receipt date. The Query payment module 1784 displays the payments, any links, and any GL entries. The operation of this use case is further described below.

Step 1. The Auditor starts the transaction.

Step 2. The Query payment module 1784 displays the licenses to which the Auditor has read access, displaying the object identifier, the Agreement ID, the Name, and the Description.

20 Step 3. The Auditor selects a license for query of payments.

Step 4. The Query payment module 1784 displays a query form that lets the Auditor enter selection conditions on the set of payments, including object identifier, Payor Name, payment Receipt Date, Amount, Withheld Amount, Late Payment Interest Amount, and/or Invoice Number.

25 Step 5. The Auditor enters query criteria and starts the query.

Step 6. The Query payment module 1784 displays the payments for the license in preferably reverse chronological order to which the Auditor has Read

access, displaying the object identifier, object identifier, Payor Name, payment Receipt Date, Amount, Withheld Amount, Late Payment Interest Amount, and/or Invoice Number.

Step 7. The Auditor selects a payment.

5           Step 8. The Query payment module 1784 displays a list of General Ledger entries, if any, showing the Entry Type, GL Account Number, Amount, Currency Unit Symbol, and Entry Date. The Query payment module 1784 preferably orders the entries by GL Account Number.

10           Step 9. The Query payment module 1784 displays a list of Payment Detail Allocations, showing the Royalty Statement Start Date Time and Amount Allocated. The Query payment module 1784 preferably orders the allocations by License Name, Start Date Time, and Line Number.

15           Step 10.       The Query payment module 1784 displays a list of License Payment Allocations, showing the License Name, Payment Number, Expected License Revenue Description and Due Date, and Amount Allocated. The Query payment module 1784 preferably orders the allocations by License OID, Term Number, and Payment Number.

Step 11.       The Auditor ends the transaction.

20           At any time, if the Data Entry Clerk presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### ***Maintain GL Accounts***

25           The maintain GL accounts use case 10702 is diagramed in FIG. 107. According to this use case, the License Administrator creates, updates, or removes accounts from the Licensing Database 1204. The operation of this use case is further described below.

Step 1. The License Administrator starts the transaction.

Step 2. The Maintain GL accounts module 1786 displays the list of current GL Accounts, displaying the Account Number and the Description.

Step 3. The License Administrator modifies the set of GL Accounts, performing any of the following actions on the set of accounts:

5                   Step 3.1.       The License Administrator creates a new GL Account, entering the Account Number and the Description.

                  Step 3.2.       The License Administrator selects one of the current GL Accounts and modifies the Account Number or the Description.

10                  Step 3.3.       The License Administrator selects one of the current GL Accounts and removes it.

                  Step 4. The License Administrator ends the transaction.

Step 5. The Maintain GL accounts module 1786 commits the changes in the Licensing Database 1204. The Maintain GL accounts module 1786 confirms the commit to the License Administrator.

15                  If a GL Account relates to any General Ledger Entries, on modifying the account number, the Maintain GL accounts module 1786 prompts the License Administrator whether to change the account number for the Entries or to cancel the transaction.

### ***Link Payment to Entity***

20                  The link payment to entity use case 15102 is diagramed in FIG. 151. This use case extends the Modify Payment use case to link or unlink the portion of a payment linked to a particular license to a specific set of named entities in the Licensing Database 1204. The License Administrator allocates some portion of the portion of a payment to a specific named entity.

25                  It will be illustrate to consider the following scenario. A licensor licenses packages of assets to licensees with a license agreement. The licensees use the assets to produce products that generate revenue. Depending on the license agreement

compensation terms, the licensee pays various kinds of payments to the licensor: fees or royalties. These payments constitute a revenue stream to the licensor as a series of payments from the licensee. A single payment may encompass more than one license, so the link between payment and license has a percent allocation that allocates a  
5 specific percentage of the payment to a particular license.

The licensor is usually an organization. Often, the licensor is not the business unit or person that ultimately "gets" or recognizes the revenue from a license for accounting purposes. That entity may be an organizational child of the licensor, but not necessarily, and there may be more than one entity that gets revenue from a  
10 license. A licensor therefore must have a way to allocate payment revenue from a particular license (a portion of a payment) to one or more named entities (people or organizations).

The operation of this use case is further described below.

Step 1. The link payment to entity module 15104 passes in the object  
15 identifier for a license agreement and a payment (a License Payment link).

Step 2. The System displays the current set of links to entities from this license-payment link, displaying the name, description, and entity type for each linked entity.

Step 3. The License Administrator uses the Query Entities use case to display  
20 a list of entities. The License Administrator selects a single named entity from the result of the query and links it to the license-payment link. The Query Entity use case returns the object identifier for the entity to this use case.

Step 4. The link payment to entity module 15104 displays a form that lets the license administrator allocate a percentage of the license-payment amount to the  
25 entity. It displays a default value of 100% if there are no license payment allocations for this license-payment link or the difference between the sum of existing license payment allocations for this license-payment link and 100% .

Step 5. The link payment to entity module 15104 creates a link between the license-payment and the named entity, inserting a row in the License Payment Allocation table and displaying the new link in the displayed list of links.

Step 6. The License Administrator repeats steps 3-6 or ends the use case and returns to the extending use case.

If the License Administrator wants to remove a given link, he or she selects it in the displayed list of links and tells the System to delete the link. The System then removes the link from the display and schedules the link for deletion from the database (the extending use case is running the transaction). The License Administrator can remove or add any number of links in this use case.

### *Display Payments*

The display payments use case 15202 is diagramed in FIG. 152. This use case displays a table of license agreements with the basic data for each agreement. The Actor may select one or more agreements and conduct further operations such as Modify License Agreement. The operation of this use case is further described below:

Step 1. The Actor starts the transaction.

Step 2. The display payments module 15204 displays a table of all the License Agreements to which the Actor has read access using the Display License Agreements use case.

Step 3. The Actor selects a single license agreement from that table.

Step 4. The display payments module 15204 displays a table of all the Payments from the Licensing Database 1204 to which the Actor has read access. Each row of the table corresponds to a single Payment. The columns include the Payor Name, the Currency Unit Symbol, the Payment Amount, the Late Payment Interest Amount, the Receipt Date, the Invoice Number, and the Payment-to-License Allocation Percent.



Step 5. The Actor ends the transaction.

This use case has a number of extensions:

- 5           °       If the Actor wants to enter a new payment, the Enter Payment use case extends this one in a separate transaction. When the extending use case ends, the display payments module 15204 displays the new payment and its fields in the table.
- °       If the Actor wants to see or modify the details of the payment, the Modify Payment use case extends this one in a separate transaction. The Actor selects a payment from the table and initiates the extending use case. The saved changes from the extending use case appear in the table when that use case ends.
- 10          °       If the Actor wants to print a payment, the Print Payment use case extends this one. The Actor selects one or more payments from the table and initiates the extending use case.
- °       If the Actor is the License Administrator and wants to remove the selected payments, the Remove Payment use case extends this one. The Actor selects one or more payments from the table and initiates the extending use case. When the extending use case returns, the display payments module 15204 removes the selected payment or payments from the table.

### *Time Period Use Cases*

20           The invention supports time period related structures. For example, Royalties and Fees often have a recurring payment. For example, a royalty may be due at the end of every quarter, on every June 15, or something similar. Most license royalties and fees call for monthly, quarterly, or annual payments.

25           Generally, recurring periods may terminate in one of two ways. A count-limited period has a certain number of periodic payments. For example, you could have a yearly fee due on June 15th for five payments. A date-limited period goes to a certain end date: a royalty paid quarterly on the 15th of the second month of the quarter ending on February 15, 2015.

The beginning interval from the start date to the first recurring date may differ significantly in size from the other dates, as may the period from the last recurring date to an end date for a date-terminated period. For example, you may owe an annual fee every June 15th, with the sequence starting on June 1. The first interval will be 15  
5 days, while the second and onwards will be one year.

In this section, time period use cases are described.

### *Enter Recurring Time Period*

The enter recurring time period use case 15602 is diagramed in FIG. 156.  
10 This use case is used by other use cases that must enter a recurring period of some kind into the Licensing Database 1204. The use case permits the actor to enter the repetition structure for the recurring period. The operation of this use case is further described below.

Step 1. The enter recurring time period module 15602 displays a data entry  
15 form containing the Time Unit (default Year), the Repetition (default 1), and the Recurring Period Type (default Count Limited).

Step 2. The Actor chooses a Time Unit, a Repetition, and/or a Recurring Period Type.

Step 3. The enter recurring time period module 15602 returns the appropriate  
20 object back to the using use case. The enter recurring time period module 15602 sets the Time Period's Time Period Type field to 'R' to indicate a recurring period object.

This use case has a number of extensions.

- ° Depending on the Time Unit, the enter recurring time period module 15602 displays a set of options that the actor can use to control complex recurring  
25 structures.

- ° If Time Unit is Year, the enter recurring time period module 15602 displays the Yearly options and allows the actor to choose between two sets of options:

-125-

- Specific Month and Specific Day: June 15th, November 2
  - Ordinal Week of Month and Day of Week and Specific Month: first Tuesday in January
- 5       ◦ If Time Unit is Quarter, the enter recurring time period module 15602 displays the Quarterly options and allows the actor to choose between three sets of options:
- Specific Day and Month: first day of second month of quarter
  - Specific Day of Week and Month: first Monday in second month of quarter
- 10       ◦ Specific Day: 45th day of quarter
- If Time Unit is Month, the enter recurring time period module 15602 displays the Monthly options and allows the actor to choose between two sets of options:
- Specific Day: 15th, 21st
- 15       ◦ Ordinal Week of Month and Day of Week: first Tuesday of the month
- If Time Unit is Week, the enter recurring time period module 15602 displays the Weekly options and allows the actor to enter the day of the week: Tuesday
- 20       ◦ If Time Unit is Day, the enter recurring time period module 15602 displays nothing.
- If the actor sets the Recurring Period Type to Count Limited, the enter recurring time period module 15602 displays the Number of Recurrences. This controls the number of recurring dates within the whole period.
- 25       ◦ If the actor sets the Recurring Period Type to Date Limited, the enter recurring time period module 15602 displays the End Date field. This controls the number of dates within the whole period by ending the whole period at a certain date.

### *Modify Recurring Time Period*

The modify recurring time period use case 16002 is diagramed in FIG. 160. This use case is used by other use cases that must modify a recurring period of some kind in the Licensing Database 1204. The use case permits the actor to modify the existing repetition structure for the recurring period or to change the type of period completely between the two concrete types (date-limited and count-limited). The operation of this use case is further described below.

Step 1. The actor passes in the object identifier for a time period to modify (:TimePeriod\_ID).

Step 2. The modify recurring time period module 16004 displays a data entry form containing the Time Unit (default Year), the Repetition (default 1), and the Recurring Period Type (default Count Limited), all displaying the current values for the time period the actor passes in (:TimePeriod\_ID) queried from the Licensing database 1204.

Step 3. The Actor changes the Time Unit, a Repetition, and/or the Recurring Period Type.

Step 4. The modify recurring time period module 16004 returns the appropriate object back to the using use case.

This use case has a number of extensions.

° If the Recurring Period Type is Count Limited, the modify recurring time period module 16004 adds the Count\_Limited\_Recurring\_Period table to the following SQL queries as the <concrete subclass> and displays the Number of Recurrences. This controls the number of recurring dates within the whole period. If the actor changes the Recurring Period Type to Date Limited, the modify recurring time period module 16004 changes to display the End Date field and deletes the count-limited row when the transaction completes, replacing it with a new Date\_Limited\_Recurring\_Period row.

- 5
  - ° If the Recurring Period Type is Date Limited, the modify recurring time period module 16004 adds the Date\_Limited\_Recurring\_Period table to the following SQL queries as the <concrete subclass> and displays the End Date field. This controls the number of recurring dates within the whole period by ending the period at a certain date. If the actor changes the Recurring Period Type to Count Limited, the modify recurring time period module 16004 changes to display the Number of Recurrences field and deletes the date-limited row when the transaction completes, replacing it with a new Count\_Limited\_Recurring\_Period row.
- 10
  - ° Depending on the Time Unit, the modify recurring time period module 16004 displays a set of options that the actor can use to control complex recurring structures.
- 15
  - ° If Time Unit is Year, the modify recurring time period module 16004 queries the Yearly options, displays them, and allows the actor to modify the options, choosing between two sets:
    - ° Specific Month and Specific Day: June 15th, November 2
    - ° Ordinal Week of Month and Day of Week and Specific Month: first Tuesday in January
- 20
  - ° If Time Unit is Quarter, the modify recurring time period module 16004 queries the Quarterly options, displays them, and allows the actor to choose between three sets:
    - ° Specific Day and Month: first day of second month of quarter
    - ° Specific Day of Week and Month: first Monday in second month of quarter
    - ° Specific Day: 45th day of quarter
- 25
  - ° If Time Unit is Month, the modify recurring time period module 16004 queries the Monthly options, displays them, and allows the actor to choose between two sets:
    - ° Specific Day: 15th, 21st

-128-

- Ordinal Week of Month and Day of Week: first Tuesday of the month
- If Time Unit is Week, the modify recurring time period module 16004 displays the Weekly options and allows the actor to enter the day of the week:  
5 Tuesday
- If Time Unit is Day, the modify recurring time period module 16004 queries and displays nothing.

### *Reports Use Cases*

10 Reports are a critical part of the Licensing module 1102. It is through reports that the license administrator and auditor (or any other user) obtain a great amount of useful information from system. All available reports are listed in a reports pane 10804 of a reports view 10802 (FIG. 108).

15 Preferably, a report engine is used to generate the reports. When the user selects a report, the licensing system 1102 generates one or more commands that collectively encapsulates the user-selected report type and any user-provided parameters. The details of these commands will be apparent to persons skilled in the relevant art(s). These commands are in the language of the report engine. The report engine generates the report pursuant to the commands. In doing so, the report engine accesses data (as indicated in the commands) in the databases discussed herein. In an  
20 embodiment, a commercial reporting module is used as the report engine, such as the commercially available Crystal Reports product.

### *Generate Report*

25 The generate report use case 10902 is diagramed in FIG. 109. The License Manager or Auditor (the "Actor") requests a particular report from a list the Generate report module 1788 displays. The Generate report module 1788 runs and displays or

prints the selected report. Preferably, to run a report on an individual object, the Print Object use case or related print use case is used. The operation of this use case is further described below.

Step 1. The Actor begins the transaction to run a report.

5           Step 2. The Generate report module 1788 constructs a list of available reports, displaying the name and description for the report in the reports view 10802 (FIG. 108).

Step 3. The Actor selects a report and requests that it be run, supplying appropriate parameters through a parameter form as requested and required, including  
10           the report destination (print or screen or file), ending the transaction.

Step 4. The Generate report module 1788 runs the report and sends it to the indicated destination.

The following are examples of reports: These are presented for illustrative purposes, and the invention is not limited to these examples. The invention is  
15           directed to include any reports of interest that can be generated from the information contained in the databases described herein. Implementation of these and other reports will be apparent to persons skilled in the relevant art(s). It is noted that license related reports can be in a plurality of forms, such as those shown in the figures, and in forms including hyperbolic trees (described above).

20           °       Payment exception report: This report tracks the payment amount allocated to a license agreement and compares it to the expected revenue amount providing a balance due. Discrepancies between the expected revenue and the allocated revenue are flagged preferably in red within parenthesis. An example payment exception report is shown in FIGS. 110A-110C.

25           °       IP Asset - Agreement Financial Detail (Summary of Intellectual Property): This report lists the license agreements for each licensed IP asset package, providing the payment amount allocated to the asset package and the expected revenue. An example IP Asset - Agreement Financial Detail (Summary of Intellectual Property) report is shown in FIGS. 111A-111D.

° Licensee Profile: This graph and detail report provides an overall view of a licensee's agreements. The graph presents total allocated payments and total expected revenue for each agreement. An example Licensee Profile report is shown in FIGS. 112A-112F.

5           ° Agreement Summary: The purpose of this report is to provide "front page" agreement and contact information and to list the compensation terms of the agreement. An example of this report is shown in FIG. 113.

          ° Licensee Asset Package Summary: The purpose of this report is to summarize which asset packages and individual assets are licensed by each licensee.  
10       An example of this report is shown in FIGS. 114A and 114B.

          ° Payment Allocation Report: This report lists the payments for each agreement and the breakout for each IP asset. An example of this report is shown in FIGS. 115A-115C.

          ° Royalty Statement Summary: This report lists the royalty statements  
15       for each agreement providing the line item details for each royalty statement. An example of this report is shown in FIG. 116.

          ° IP Asset - Historical Royalties: This report for each asset in a licensed package lists the agreements that license out the asset and provides a running total of royalties earned. An example of this report is shown in FIG. 117.

20       ° The Summary of Intellectual Property Report relates an IP asset to its overall licensing revenue.

          ° The Intellectual Property Licensee Summary Report provides a summary of agreements for each licensee.

25       ° The Royalty Expense Allocation Report tracks royalty earnings posted in the General Ledger but not yet verified.

          ° The Licensee Historical Earned Royalty Graph shows Guaranteed Minimum, Actual Payment, Expensed payment, and Term Required totals for a specified Licensee by time period (date range or audit period selection).



-131-

° The IP Historical Earned Royalty Graph shows G/L total and Expensed total for an IP Asset.

° The Overdue Payment Graph shows Statement Received dates, Term total by due date, Payment total by payment date with some sort of flag and value for overdue interest owed or paid.

° The Draft License is a report that is in the form of a license agreement document, that is generated by the system based on information in the databases. This draft document can be modified as necessary to produce the actual license agreement document.

10

### *Security Use Cases*

In an embodiment, Licensing system security includes the security features described above. In other embodiments, Licensing system security includes any combination of the above with one or more other features, such as the following administrative use cases, the ability to assign a security classification in the various document and payment entry and modification use cases, and the ability to change privileges on asset groups.

15

### *Administer Entities (Users)*

The administer users use case 11802 is diagramed in FIG. 118. According to this use case, the system Administrator creates, modifies, and/or removes users from the core database 1208.

20

Step 1. The System Administrator starts the transaction.

Step 2. The Administer entities module 1715 displays a list of users ordered by name, displaying the User Name, User Full Name, and Password (the password field appears but the actual password does not); there is also a Confirm Password for validating password entry.

25

Step 3. The System Administer takes one of the following actions:

Step 3.1. The System Administrator creates a new User, entering the User Name, the Full\_Name, and the Password (twice) (the Administer entities module 1715 generates the User ID). The Administer entities module 1715 encrypts the password.

Step 3.2. The System Administrator selects one of the current Users and modifies the Name, the Full Name, or the Password.

Step 3.3. The System Administrator selects one of the current Users and removes it.

Step 4. The System Administrator ends the transaction.

Step 5. The Administer entities module 1715 commits the changes in the core database 1208. The Administer entities module 1715 confirms the commit to the System Administrator.

This use case has a number of extensions:

° If the System Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

° If removing a User, the Administer entities module 1715 deletes the information from the Licensing database 1204 and moves it to the Recycle Bin (implementation may differ regarding details on deferring the actual SQL deletes, and there is no notification of commit to the System Administrator). The Administer entities module 1715 confirms the commit to the System Administrator. At any time until clearing the Recycle Bin, the administrator can restore the removed User and its related information.

### *Administer Security Classes*

The administer security classes use case 11902 is diagramed in FIG. 119. According to this use case, the system Administrator sees a list of all security classes. The system Administrator can add a new class to the list, can modify the name and

access control list of a class, or can remove a class. This use case is further described below.

Step 1. The System Administrator starts the transaction.

5 Step 2. The Administer security classes module 1717 displays a list of Security Classes, displaying the Security Class ID and Name, and a list of the Entities, displaying the Entity ID and User Name.

Step 3. The System Administer takes one of the following actions:

10 Step 3.1. The System Administrator creates a new Security Class, entering the Name (the Administer security classes module 1717 generates the Security Class ID), and requests that the IPAM Security Subsystem create the object.

Step 3.2. The System Administrator selects one of the current Classes and modifies the Name.

Step 3.3. The System Administrator selects one of the current Security Classes and requests that the IPAM Security system 1602 remove it.

15 Step 3.4. The System Administrator selects a Security Class. The Administer security classes module 1717 displays the permissions for that Security Class for each Entity. The System Administrator then modifies the permissions for any entity, including those with no permissions. The System Administrator requests the IPAM Security Subsystem 1602 to set the permissions for the modified Security Class and Entities.

Step 4. The System Administrator ends the transaction.

Step 5. The Administer security classes module 1717 commits the changes in the Licensing Database 1204. The Administer security classes module 1717 confirms the commit to the System Administrator.

25 At any time, if the System Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

### *Grant Permissions*

5 The grant permissions use case 12002 is diagramed in FIG. 120. According to this use case, the System Administrator selects a secure object (an asset package or classifier), selects an entity (user or user group), and grants read, write, and/or delete permission to the entity for the object. The grant permissions module 1719 makes the permissions permanent through the IPAM Security Subsystem 1602. This use case is further described below.

Step 1. The System Administrator starts the transaction.

10 Step 2. The Grant permissions module 1719 displays a list of the secure objects to which the System Administrator has access based on their current authentication user name. It gets this from the IPAM Security Subsystem 1602. The Grant permissions module 1719 displays the objects sorted into two groups, asset packages and classifiers, distinguishing them by type (an icon, separate panes, or whatever). The asset packages display the Package OID and the Name. The classifiers  
15 display the Security Class OID and the Name.

Step 3. The System Administrator selects a secure object.

Step 4. The Grant permissions module 1719 displays a control that lists available entities ordered by entity type and name, which it gets through the IPAM Security Subsystem 1602.

20 Step 5. The System Administrator selects one or more entities.

Step 6. The Grant permissions module 1719 displays the current permission settings on the secure objects (Read, Write, Delete), distinguishing settings that are the same for all selected secure objects and entities from those that are different (for example, a greyed check mark).

25 Step 7. The System Administrator sets the permission flags (Read, Write, Delete) to new values as required, then ends the transaction.

Step 8. The Grant permissions module 1719 passes the changes to the IPAM Security Subsystem 1602, which updates the security tables in the Core Database 1208.

### *Administer Roles*

5           The administer roles use case 12402 is diagramed in FIG. 124. According to this use case, the System Administrator creates, modifies, and/or removes roles from the Licensing database 1204. The operation of this use case is further described below.

Step 1. The System Administrator starts the transaction.

10           Step 2. The Administer roles module 1721 displays a list of Roles, displaying the Role ID, Name, and Description.

Step 3. The System Administer takes one of the following actions:

15           Step 3.1.     The System Administrator creates a new Role, entering the Name and Description (the Administer roles module 1721 generates the Role ID)

Step 3.2.     The System Administrator selects one of the current Roles and modifies the Name or the Description.

Step 3.3.     The System Administrator selects one of the current Roles and removes it.

20           Step 4. The System Administrator ends the transaction.

Step 5. The Administer roles module 1721 commits the changes in the Licensing database 1204. The Administer roles module 1721 confirms the commit to the System Administrator.

This use case has a number of extensions:

25           °     If the System Administrator presses F1 or the equivalent, the application extends the use case with the Display Help use case.

- If removing a Role, the Administer roles module 1721 deletes the information from the licensing database 1204 and moves it to the Recycle Bin (implementation may differ regarding details on deferring the actual SQL deletes, and there is no notification of commit to the System Administrator). The Administer roles module 1721 confirms the commit to the System Administrator. At any time until clearing the Recycle Bin, the administrator can restore the removed Role and its related information.

### *Currency Use Cases*

The various monetary amounts in Licensing module 1102 may be in any currency. This can cause problems when payments, royalty statements, and compensation terms list the amounts in different currencies. The Convert Currency use case provides a currency conversion mechanism for the generate report module 1788. This is visible to the user when the user takes an action that needs to operate on amounts in different currencies.

Currency conversion depends on a database of conversion ratios. These ratios vary over time. Depending on the customer's needs, the customer could want to have a different ratio every day, every month, or every year. Or, they could just want a single conversion rate. The Maintain Currency Conversions use case corresponds to a menu entry that lets the License Administrator enter open-ended time intervals and conversion ratios that hold during those intervals.

### *Convert Currency*

The convert currency use case 12202 is diagramed in FIG. 122. This use case is used by several other use cases to convert monetary amounts. The User passes the monetary amount, the currency, and the date to the use case. The Convert currency module 1790 converts the currency using the currency conversion intervals from the

Licensing database 1204 and returns the dollar amount. The operation of this use case is further described below.

Step 1. The using use case passes in the monetary amount, the currency object identifier, and the date of the transaction.

5           Step 2. The Convert currency module 1790 looks up the currency conversion rate in the licensing database 1204. The conversion rate depends on the date of the transaction, because currency rates fluctuate with time.

Step 3. The Convert currency module 1790 returns the converted amount in dollars.

10           This use case has a number of extensions:

- °           If the currency doesn't exist, the Convert currency module 1790 displays an error message: "Currency <oid> doesn't exist."

- °           If there are no currency intervals for the currency with a date less than or equal to the transaction date, the Convert currency module 1790 displays an error message: "No conversions for currency as of <Transaction Date>".

15

### *Maintain Currency Conversions*

The maintain currency conversions use case 12302 is diagramed in FIG. 123. According to this use case, the License Administrator performs various currency maintenance functions:

- 20           °           Adding a new currency
- °           Modifying an existing currency
- °           Removing a currency
- °           Adding a new currency conversion interval
- °           Removing an existing interval
- 25           °           Modifying an existing interval

The operation of this use case is further described below.

Step 1. The License Administrator begins the transaction.

Step 2. The Maintain currency use case 12302 displays a list of currencies, displaying the Unit Name, Country, and Unit Symbol for the currency:

Step 3. The License Administrator selects a currency or adds a new currency. The License Administrator optionally changes the Unit Name, Country, or Unit Symbol or removes the currency.

Step 4. If the currency is new, the Maintain currency use case 12302 creates a new currency object identifier. The Maintain currency use case 12302 displays the time interval layout of the currency, displaying the Start Date, the Dollar Conversion Rate, and the Description, if any:

Step 5. The License Administrator selects an interval, if there are any.

Step 6. The License Administrator removes the selected interval or changes the Start Date, Dollar Conversion Rate, or Description fields.

Step 7. The License Administrator repeats steps 3-6 or commits the transaction.

Step 8. The Maintain currency use case 12302 inserts any new currencies, updates any changed currencies, and deletes any removed currencies (including any currency intervals). The Maintain currency use case 12302 inserts any new intervals, updates any changed intervals, and deletes any removed intervals. The Maintain currency use case 12302 confirms the commit to the License Administrator.

This use case has an extension. Specifically, if the License Administrator wants to add a new interval to the existing set, the Maintain currency use case 12302 displays the new interval at the bottom of the list of current intervals. The License Administrator enters the Start Date, the Description, and the Dollar Conversion Rate. When the License Administrator selects another interval, the Maintain currency use case 12302 sorts the list and puts the new interval in its proper place in the list ordered by date.



### *Top Level Operational Examples Of the Licensing System*

An example thread of operation through the licensing system 1102 shall now be described with reference to a flowchart 12502 in FIG. 125. It should be understood that the present invention, including the licensing system 1102, is very powerful, very flexible, very adaptable, and very user-responsive. According to the invention, users can employ and/or adapt the functions, modules, components, objects, etc., describe herein to pursue their own strategies, goals, needs, requirements, desires, etc. Users are not limited to the particular examples described herein, such as the example in FIG. 125. Accordingly, it should be understood that the example of FIG. 125 is provided for purposes of illustration, and is not limiting.

In the following, the steps are said to be performed by a "user." Generally, this user can be any person desiring to use the licensing system 1102, including any of the actors described herein. Also, different users can perform different steps.

In step 12504, the user defines entities. These entities will later be assigned to roles, such as licensee and licensor. See, *inter alia*, the Administer Entity use case.

In step 12506, the user creates IP (intellectual property) assets. This involves entering patent assets if the licensing system 1102 is not integrated with the IPAM database 216. If the licensing system 1102 is integrated with the IPAM database 216, then there is no need to enter patent assets as such patent assets are accessible from the IPAM database 216 (although in some embodiments, the user is allowed to enter patent assets even when the licensing system 1102 is integrated with the IPAM database 216, since, for example, the IPAM database 216 may not include all U.S. and/or foreign patents, reissues, reexaminations, etc.). This step also includes entering trademark assets, entering trade secret assets, entering copyright assets, entering know how assets, etc. See, *inter alia*, the Enter Patent, Enter Trademark, Enter Copyright, Enter Trade Secret, and Enter Know How use cases.

In step 12508, the user creates asset packages. Each asset package includes IP assets to license. There are three types of asset packages: standard, group, and

descriptive. When entering a standard asset package, the user manually selects assets for inclusion into the asset package (this can be done via drag and drop operations, for example). When entering a group asset package, the user selects an IPAM group. The assets in the IPAM group become the contents of the asset package. When  
5 entering a descriptive asset package, the user provides a description of the assets being licensed, such as "All patents and patent applications currently existing as of June 18, 1997." See, *inter alia*, the Create IP Asset Package use case.

In step 12510, optionally, the user provides a revenue allocation percentage for each asset in an asset package. An asset's revenue allocation percentage within  
10 an asset package indicates the degree to which that asset contributed to license revenue attributed to the asset package. For example, if an asset within an asset package has an allocation percentage of 65%, and a license revenue of \$100 is attributed to the asset package, then that asset is considered to have been responsible for generating \$65 of the \$100 license revenue. See, *inter alia*, the Create IP Asset  
15 Package use case.

In step 12512, the user creates a license agreement. See, *inter alia*, the Enter License Agreement use case. In doing so, the user specifies the parties to the agreement, such as the licensee(s) and the licensor(s). These parties are entities who were defined in step 12506. See, *inter alia*, the Link to Party use case.

The license agreement is an instrument that operates to license IP assets from the licensor(s) to the licensee(s). According to an embodiment of the invention, the IP assets that are being licensed via the license agreement are specified via asset packages. Specifically, in step 12512, the user identifies one or more asset packages that contain the IP assets that are desired to be licensed via the license agreement.  
20 These identified asset packages are linked to the license agreement. See, *inter alia*, the Link to Asset Package use case.

A license agreement includes a number of terms agreed upon by the parties, such as whether the license is limited to particular territories, whether the license is exclusive or not exclusive, whether the license includes grant back provisions, etc.

Importantly, a license also includes compensation terms, which specify the compensation that the licensee pays to the licensor for the benefit of receiving the license to the IP assets contained in the linked asset packages. Accordingly, also in step 12512, the user enters compensation terms for the license. See, *inter alia*, the Enter Compensation Term use case.

In step 12514, optionally, the user determines the expected revenue of the license agreement. This is preferably done on a per compensation term basis. Specifically, for each compensation term, a schedule of expected future revenue payment(s) is calculated, based on the type and characteristics of the term. For example, if Term1 is a lump sum payment of \$100 to be paid on January 1, 2000, then the expected revenue of Term 1 is calculated to be \$100 to be received by the licensor on January 1, 2000. If Term2 is an ongoing royalty based on per unit sales, where the royalty rate is \$1 per 100 units and the estimated unit sales is 1000 per period (specified via the Enter Compensation Term use case), then the expected revenue of Term2 is calculated to be \$10 per period. These expected future revenue payments are displayed in the expected revenue tab 9608 of the payment dialog 9602 (FIG. 96). See, *inter alia*, the Enter Compensation Term and Create Expected Revenue use cases.

In step 12516, the user receives a royalty statement related to the license from the licensee. Licensees are typically required to send such royalty statements to the licensor, such as on a quarterly basis. The user enters the royalty statement into the system. This includes entering the period to which the royalty statement applies. This also includes entering the details of the royalty statement, where each detail corresponds to a product, the number of units of the product sold during the period, the revenue generated by the sales, and the royalty due. This information is obtained from the royalty statement. See, *inter alia*, the Enter Royalty Statement use case.

In step 12518, the user receives a license revenue payment related to the license. The user enters the payment into the system. This includes entering the amount of the payment. Optionally, this also includes allocating the payment to terms

of the license. See FIG. 98. For example, assume that a license has terms Term1 and Term2, and that in step 12514 it was calculated that on a per period basis the expected revenue of Term1 is \$100, and the expected revenue of Term2 is \$200. Assume that the payment is for \$275. In step 12518, the user may elect to allocate \$100 of the \$275 payment to Term1, and the remaining \$175 of the \$275 payment to Term2. In this example, the expected revenue of Term1 is fully satisfied by the payment, but the expected revenue of Term2 is short by \$25. This is an example of how the invention can aid the user in tracking license payments.

Optionally, step 12518 also includes allocating the payment to details of royalty statements received by the licensor from the licensee. See FIG. 97. For example, suppose that a royalty statement having details Detail1 and Detail2 are associated with the license agreement. Detail1 has an expected royalty of \$125, and Detail2 has an expected royalty of \$250. Assume that the payment is for \$275. In step 12518, the user can elect to allocate \$125 of the \$275 payment to Detail1, and the remaining \$150 of the \$275 payment to Detail2. In this example, the expected royalty of Detail1 is fully satisfied by the payment, but the expected royalty of Detail2 is short by \$100. This is another example of how the invention can aid the user in tracking license payments.

With regard to the operation of step 12518, see, *inter alia*, the Enter Payment, Modify Payment, Link to Expected Revenue, and Link to Detail use cases.

In step 12520, the user runs reports based on his needs, requirements, interests, etc. Examples of reports supported by the invention include, but are not limited to: compare payments to royalty statement details; compare payments to expected revenue; determine the revenues (payments) attributable to each asset in an asset package; and any other report discussed herein. It is noted that the invention is not limited to the reports discussed herein. Any report, particularly any financial related report, relating to the subject matter discussed herein and/or derivable from the data discussed herein is contemplated by the invention. Implementation of such reports will be apparent to persons skilled in the relevant art(s).

*Conclusion*

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not  
5 be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

***What Is Claimed Is:***

1. A computer based method of managing intellectual property (IP) related transactions, comprising the steps of:

- 5 (1) accessing a database comprising information representative of at least one IP asset;
- (2) accessing a database comprising information representative of at least one IP asset package each comprising one or more of said at least one IP asset;
- 10 (3) accessing a database comprising information representative of at least one license agreement each associated with one or more of said at least one IP asset package; and
- (4) enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, (b) said at least one IP asset package, and (c) said at least one license agreement.

2. The method of claim 1, further comprising the steps of:

- 15 (5) accessing a database comprising information representative of at least one royalty statement each associated with one or more of said at least one license agreement;
- (6) accessing a database comprising information representative of at least one payment each associated with one or more of said at least one license agreement;
- 20 and
- (7) enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, (b) said at least one IP asset package, (c) said at least one license agreement, (d) said at least one royalty statement, and (e) said at least one payment.

25 3. The method of claim 1, further comprising the step of:

(8) accessing a database comprising information representative of entities and entity roles;

wherein said at least one license agreement specifies one or more of said entities as parties to said at least one license agreement.

- 5           4.       The method of claim 1, wherein said at least one IP asset comprises at least one of: (a) a patent asset, (b) a copyright asset, (c) a trade mark asset, (d) a trade secret asset, and (e) a know how asset.
5.       The method of claim 1, wherein said at least one IP asset package comprises at least one of: (a) a standard IP asset package, (b) a group IP asset package, and (c) a descriptive IP asset package.
- 10           6.       The method of claim 5, wherein said standard IP asset package comprises any user selected combination of said at least one IP asset.
7.       The method of claim 5, wherein said group IP asset package is associated with a group, said group IP asset package comprising any assets in said group.
- 15           8.       The method of claim 7, wherein said group includes one or more patents, and wherein said group IP asset package comprises said one or more patents.
9.       The method of claim 5, wherein said descriptive IP asset package is associated with a description, and wherein said descriptive IP asset package comprises assets satisfying said description.
- 20           10.      The method of claim 1, wherein said at least one IP asset package comprises a revenue allocation percentage for each of said at least one IP asset in said at least one IP asset package.

11. The method of claim 1, wherein said at least one license agreement comprises one or more user specified compensation terms.
12. The method of claim 11, wherein said at least one license agreement further comprises information pertaining to expected revenue of said at least one license agreement.
13. The method of claim 12, wherein said expected revenue is automatically calculated for said one or more user specified compensation terms.
14. The method of claim 2, wherein said at least one royalty statement comprises information pertaining to one or more royalty statement details, wherein each of said one or more royalty statement details corresponds to a product and comprises information indicative of a number of units of the product sold during a period of said at least one royalty statement, revenue generated by the sales, and royalty due.
15. The method of claim 2, wherein said at least one payment is allocated to one or more terms of said at least one license agreement.
16. The method of claim 2, wherein said at least one payment is allocated to one or more details of one or more royalty statements associated with said at least one license agreement.
17. The method of claim 2, wherein step (4) comprises the steps of:
- (i) comparing any payment amounts allocated to said at least one license agreement to any expected revenue amounts of said at least one license agreement; and
  - (ii) generating a payment exception report based on results of step (i).



18. The method of claim 2, wherein step (4) comprises the step of:
- (i) generating an IP asset report listing any license agreements involving said at least one IP asset package, wherein the IP asset report includes information indicative of any payment amounts allocated to said at least one IP asset package, and any expected revenue of said at least one IP asset package.
19. The method of claim 2, wherein step (4) comprises the step of:
- (i) generating a licensee profile report comprising information indicative of total allocated payments and total expected revenue for each of said at least one license agreement.
20. The method of claim 2, wherein step (4) comprises the step of:
- (i) generating an agreement summary report comprising information indicative of at least contact information and compensation terms of said at least one license agreement.
21. The method of claim 2, wherein step (4) comprises the step of:
- (i) generating a licensee asset package summary report comprising information indicative of any asset packages and any IP assets licensed to each licensee.
22. The method of claim 2, wherein step (4) comprises the step of:
- (i) generating a payment allocation report comprising information indicative of any payments associated with said at least one license agreement, and allocation of said any payments to terms of said at least one license agreement.
23. The method of claim 2, wherein step (4) comprises the step of:

(i) generating a royalty statement summary report comprising information indicative of any royalty statements, and details of said any royalty statements, associated with said at least one license agreement.

24. The method of claim 2, wherein step (4) comprises the step of:

5 (i) generating a historical royalties report listing any license agreements that license an IP asset, and comprising information indicative of royalties earned for said IP asset per license agreement.

25. The method of claim 2, wherein step (4) comprises the step of:

10 (i) generating a summary of IP report comprising information indicative of overall licensing revenue of an IP asset.

26. The method of claim 2, wherein step (4) comprises the step of:

(i) generating an IP license summary report that provides a summary of license agreements for each licensee.

27. The method of claim 2, wherein step (4) comprises the step of:

15 (i) generating a royalty expense allocation report that tracks royalty earnings posted in a general ledger but not yet verified.

28. The method of claim 2, wherein step (4) comprises the step of:

20 (i) generating a licensee historical earned royalty report comprising information indicative of Guaranteed Minimum, Actual Payment, Expensed payment, and Term Required totals for said at least one license agreement.

29. The method of claim 2, wherein step (4) comprises the step of:

(i) generating an IP historical earned royalty group comprising information indicative of G/L total and Expensed total for an IP Asset.

30. The method of claim 2, wherein step (4) comprises the step of:

(i) generating an Overdue Payment report comprising information indicative of Statement Received dates, Term total by due date, and Payment total by payment date.

5 31. The method of claim 1, wherein said at least one license agreement comprises one or more terms related to at least one of territorial restrictions and field-of-use restrictions.

10 32. The method of claim 11, wherein said one or more user specified compensation terms comprises one or more of an ongoing royalty per unit term, royalty per sales term, minimum guarantee term, advance term, ongoing fee term, and lump sum fee term.

33. The method of claim 11, wherein said one or more user specified compensation terms comprises at least one term that includes a recurring payment.

15 34. A computer based method of managing intellectual property (IP) related transactions, comprising the steps of:

(1) accessing a database comprising information representative of at least one IP asset;

20 (2) accessing a database comprising information representative of at least one license agreement each associated with one or more of said at least one IP asset; and

(3) enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, and (b) said at least one license agreement.

35. The method of claim 34, further comprising the steps of:

(4) accessing a database comprising information representative of at least one royalty statement each associated with one or more of said at least one license agreement;

5 (5) accessing a database comprising information representative of at least one payment each associated with one or more of said at least one license agreement; and

(6) enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, (b) said at least one license agreement; (c) said at least one royalty statement; and (d) at least one  
10 payment.

36. A system for managing intellectual property (IP) related transactions, comprising:

means for accessing a database comprising information representative of at least one IP asset;

15 means for accessing a database comprising information representative of at least one license agreement each associated with one or more of said at least one IP asset; and

means for enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, and (b) said  
20 at least one license agreement.

37. The system of claim 36, further comprising:

means for accessing a database comprising information representative of at least one royalty statement each associated with one or more of said at least one license agreement;

25 means for accessing a database comprising information representative of at least one payment each associated with one or more of said at least one license agreement; and

means for enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, (b) said at least one license agreement; (c) said at least one royalty statement; and (d) at least one payment.

5           38.     A computer program product comprising control logic stored in a computer useable medium, said control logic comprising:

              means for enabling a computer to access a database comprising information representative of at least one IP asset;

              means for enabling a computer to access a database comprising information  
10           representative of at least one license agreement each associated with one or more of said at least one IP asset; and

              means for enabling a computer to enable processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, and (b) said at least one license agreement.

15           39.     The computer program product of claim 38, wherein said control logic further comprises:

              means for enabling a computer to access a database comprising information representative of at least one royalty statement each associated with one or more of said at least one license agreement;

20           means for enabling a computer to access a database comprising information representative of at least one payment each associated with one or more of said at least one license agreement; and

              means for enabling a computer to enable processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one  
25           IP asset, (b) said at least one license agreement; (c) said at least one royalty statement; and (d) at least one payment.

40. A method of managing intellectual property (IP) related transactions, comprising the steps of:

(1) enabling a user to enter information representative of at least one IP asset;

5 (2) enabling a user to enter information representative of at least one license agreement each associated with one or more of said at least one IP asset; and

(3) enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, and (b) said at least one license agreement.

10 41. The method of claim 40, further comprising the steps of:

(4) enabling a user to enter information representative of at least one royalty statement each associated with one or more of said at least one license agreement;

15 (5) enabling a user to enter information representative of at least one payment each associated with one or more of said at least one license agreement; and

(6) enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, (b) said at least one license agreement, (c) said at least one royalty statement, and (d) said at least one payment.

20 42. A system for managing intellectual property (IP) related transactions, comprising:

means for enabling a user to enter information representative of at least one IP asset;

25 means for enabling a user to enter information representative of at least one license agreement each associated with one or more of said at least one IP asset; and

means for enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, and (b) said at least one license agreement.

43. The system of claim 42, further comprising:

5 means for enabling a user to enter information representative of at least one royalty statement each associated with one or more of said at least one license agreement;

means for enabling a user to enter information representative of at least one payment each associated with one or more of said at least one license agreement; and

10 means for enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, (b) said at least one license agreement, (c) said at least one royalty statement, and (d) said at least one payment.

44. A computer program product comprising control logic stored in a computer useable medium, said control logic comprising:

15 means for enabling a computer to enable a user to enter information representative of at least one IP asset;

means for enabling a user to enter information representative of at least one license agreement each associated with one or more of said at least one IP asset; and

20 means for enabling processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, and (b) said at least one license agreement.

45. The computer program product of claim 44, wherein said control logic further comprises:

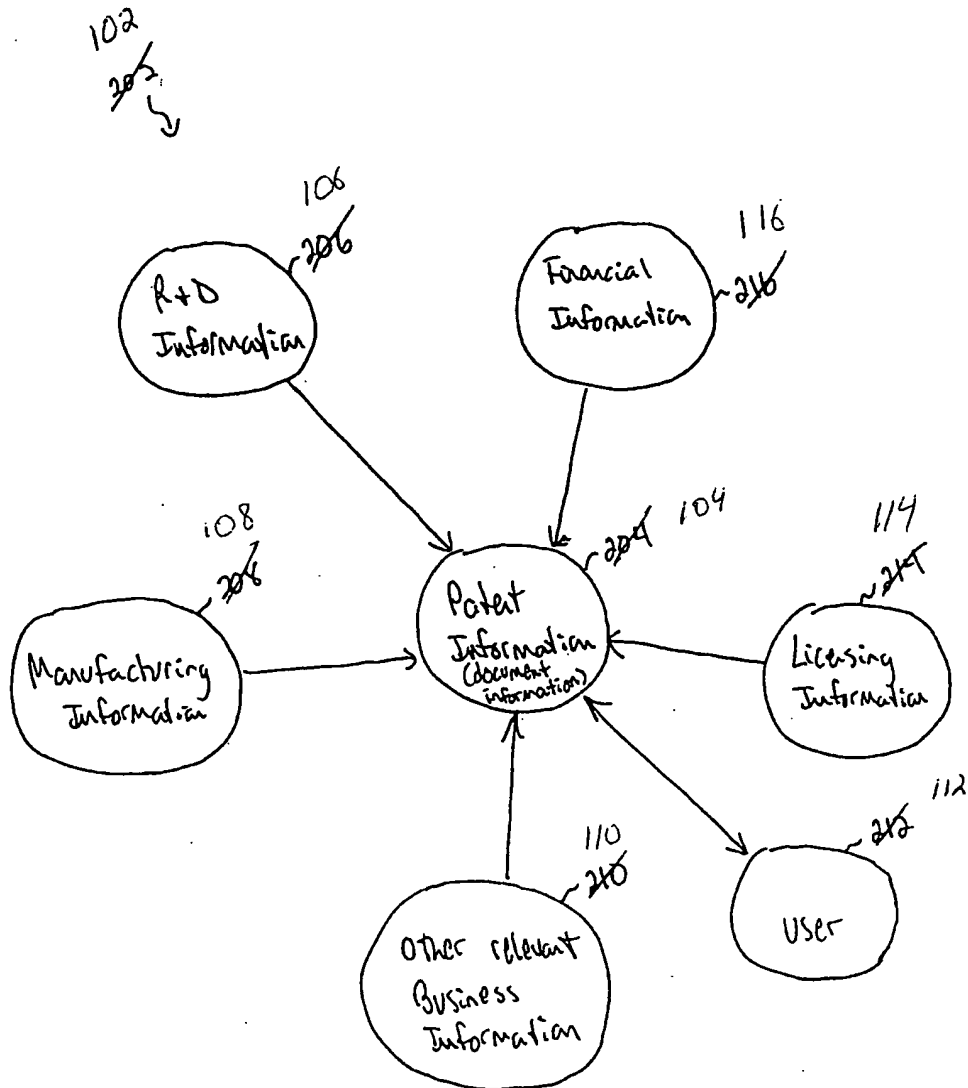
means for enabling a computer to enable a user to enter information representative of at least one royalty statement each associated with one or more of said at least one license agreement;

5 means for enabling a computer to enable a user to enter information representative of at least one payment each associated with one or more of said at least one license agreement; and

means for enabling a computer to enable processing of, in a manner specified by a user command, information representative of at least one of: (a) said at least one IP asset, (b) said at least one license agreement, (c) said at least one royalty statement,  
10 and (d) said at least one payment.



1/193

FIG. ~~2~~ 1

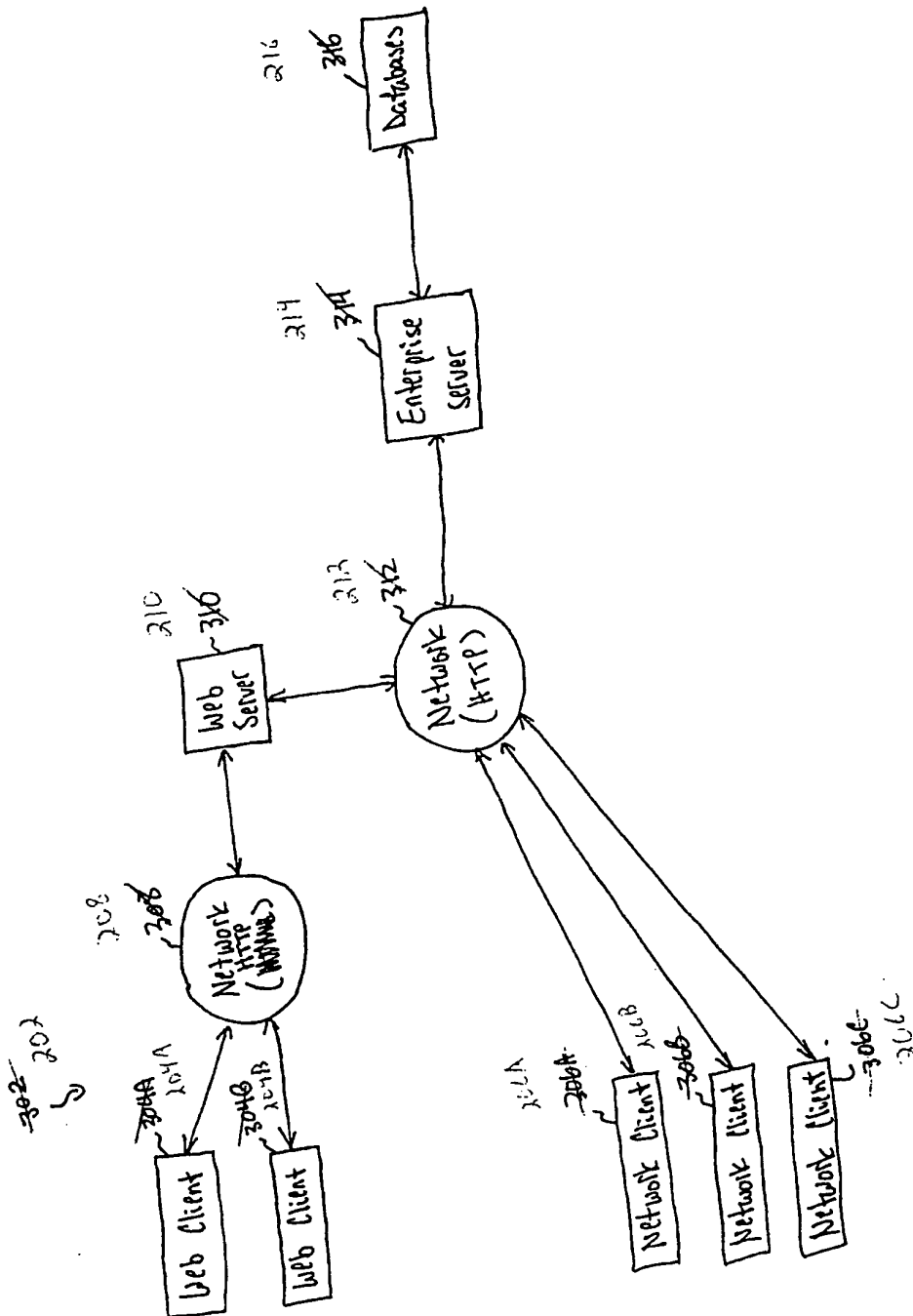
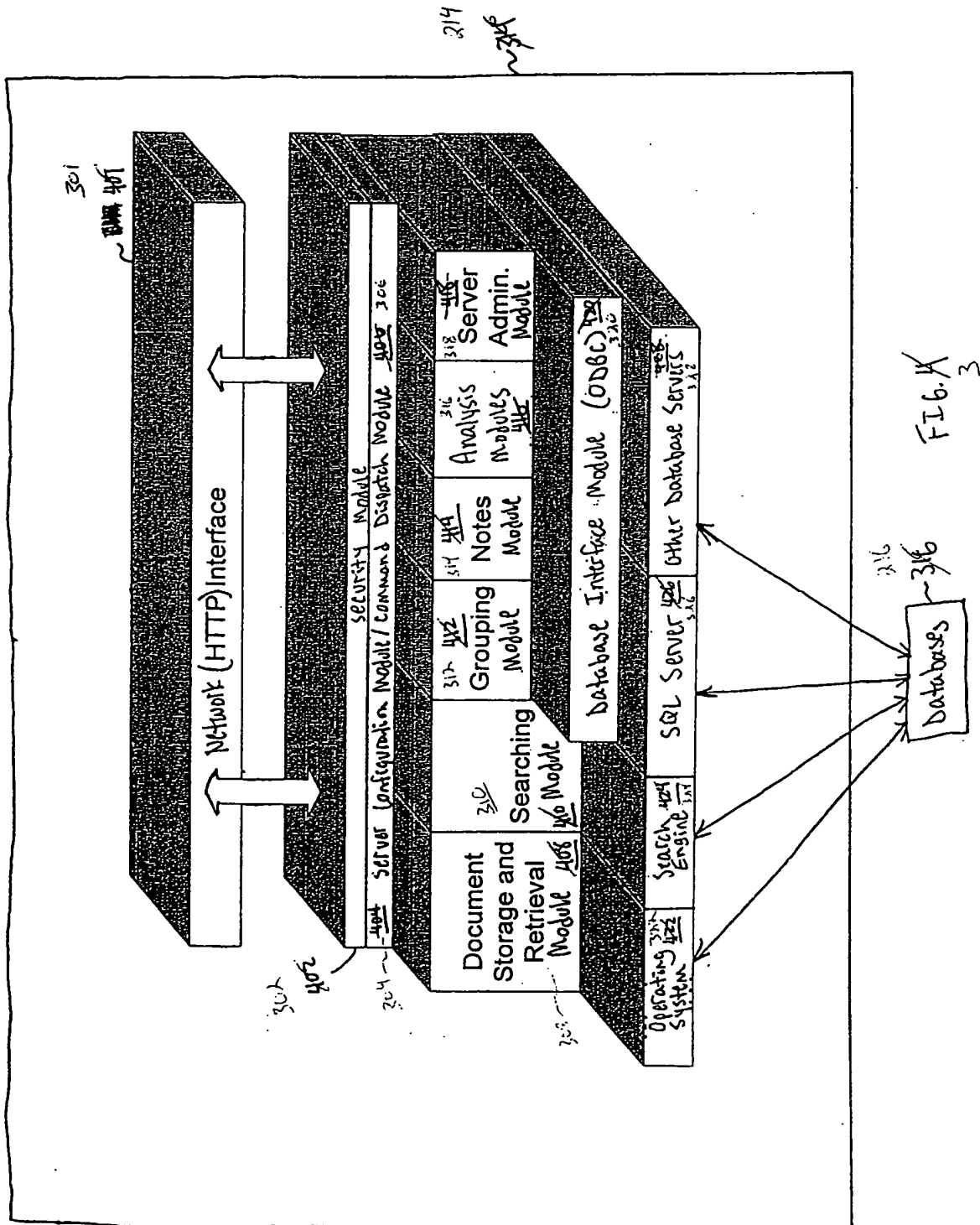
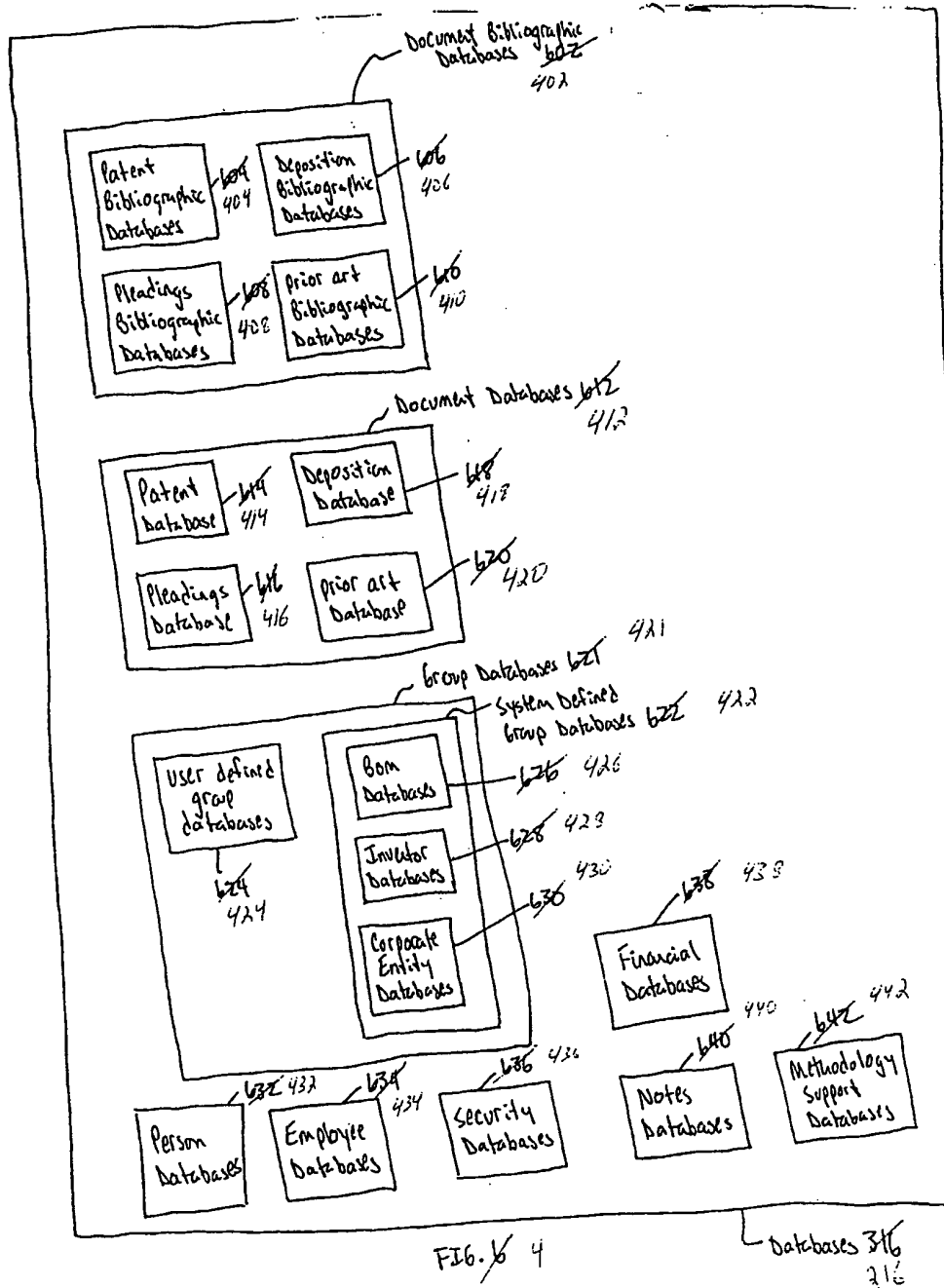


FIG. 2

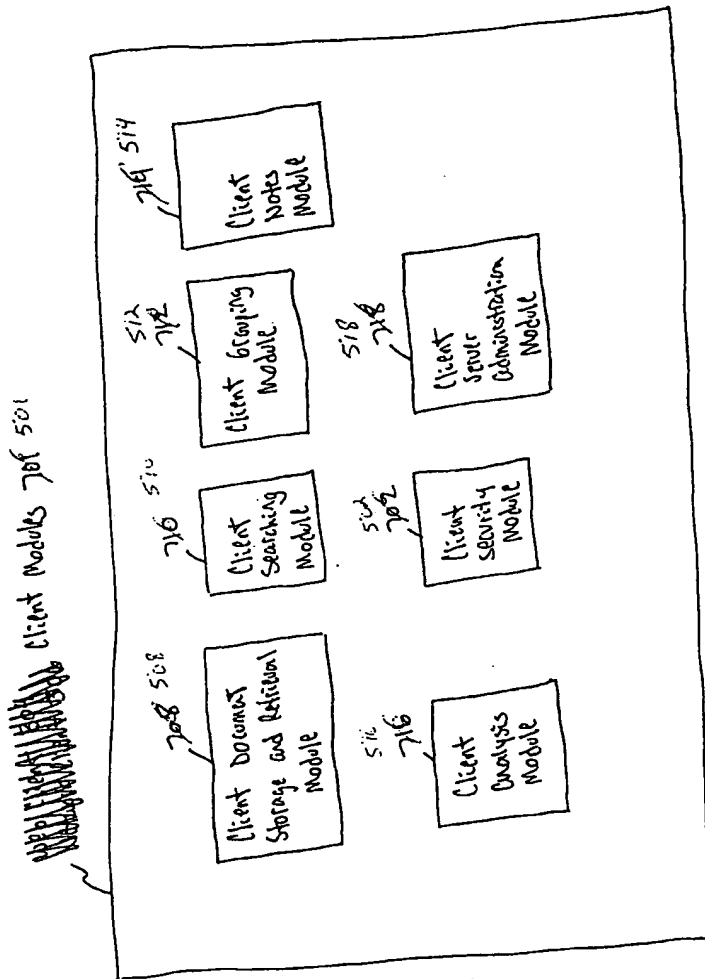
3/193



4/193



5/193



5  
FIG. 7A

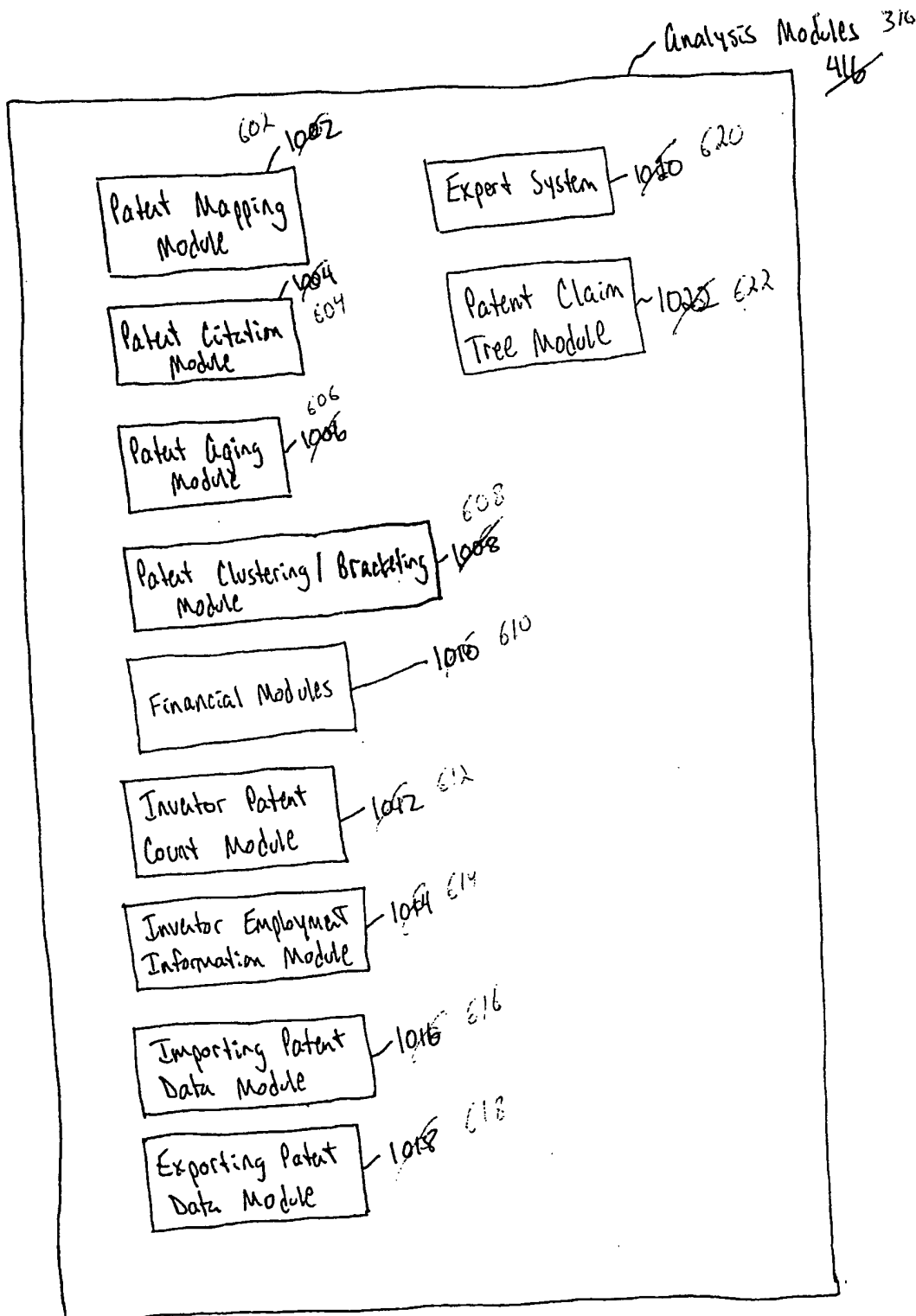
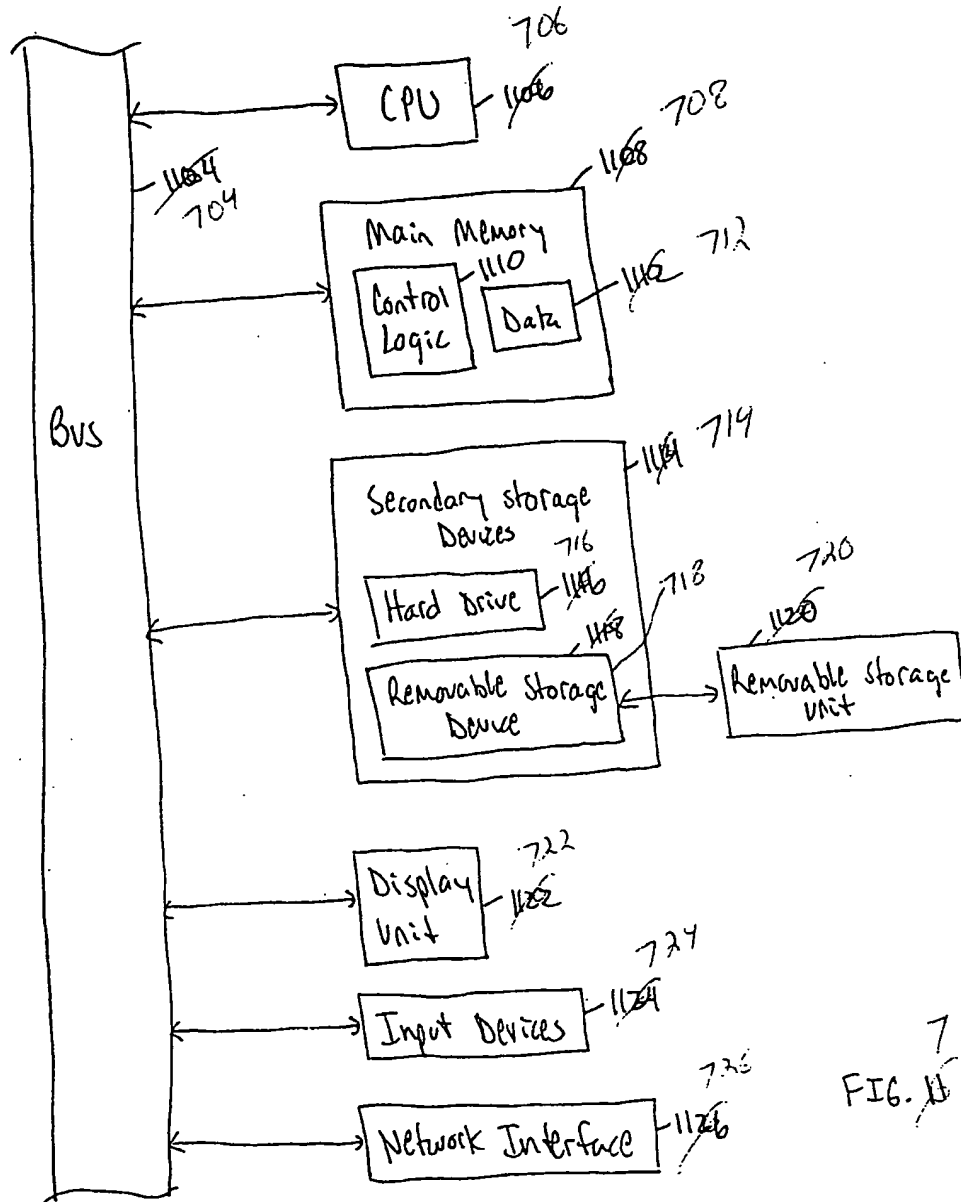


FIG. 10

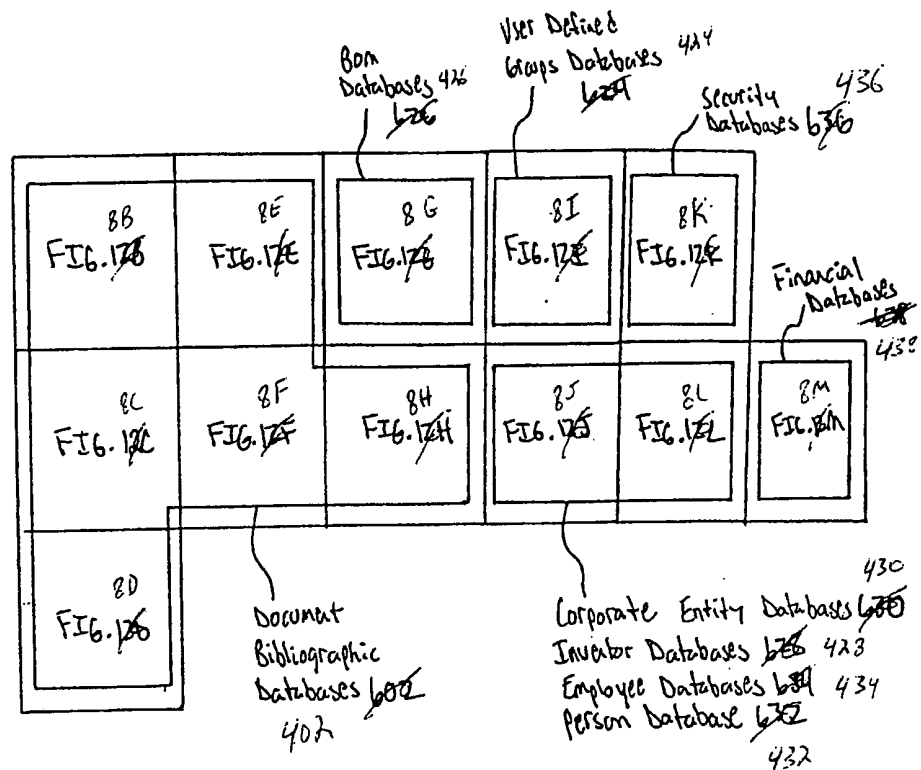
7/193

702  
1102  
↓



7  
FIG. 15

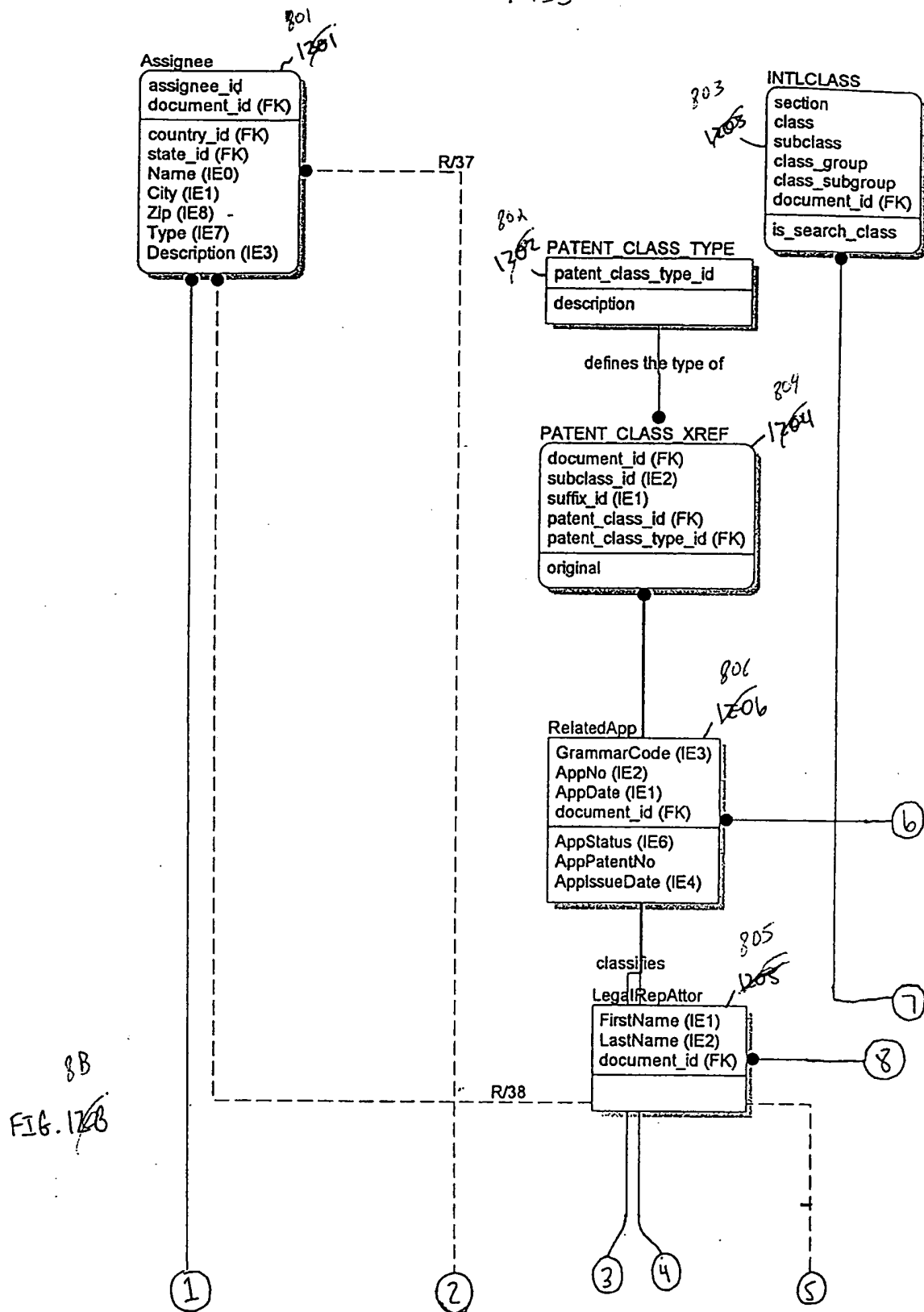
8/193



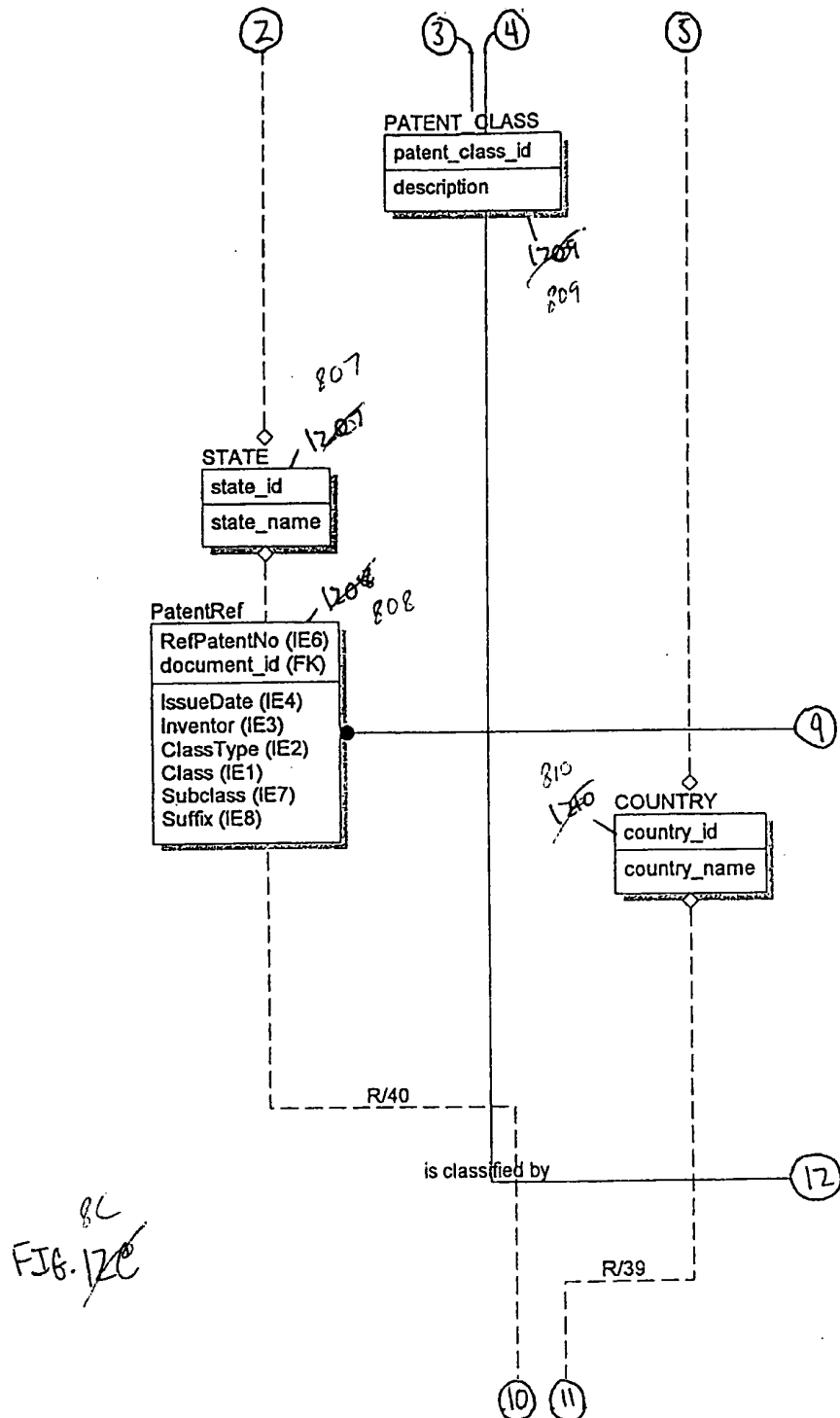
8A  
FIG. 124



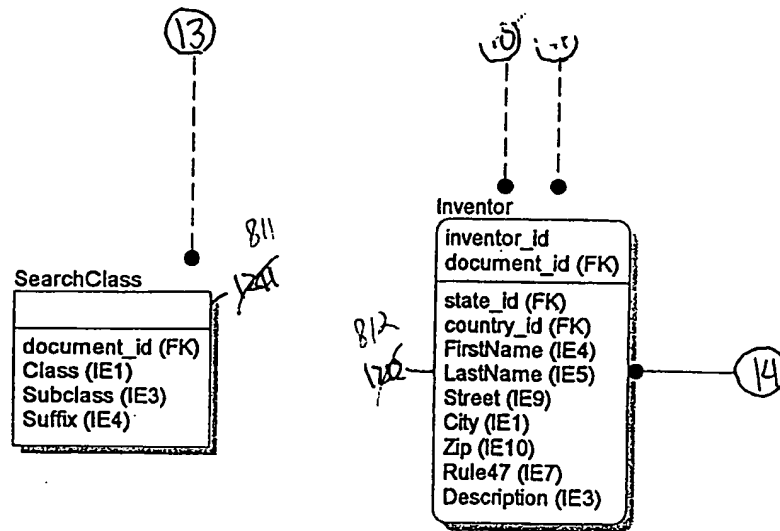
9/193



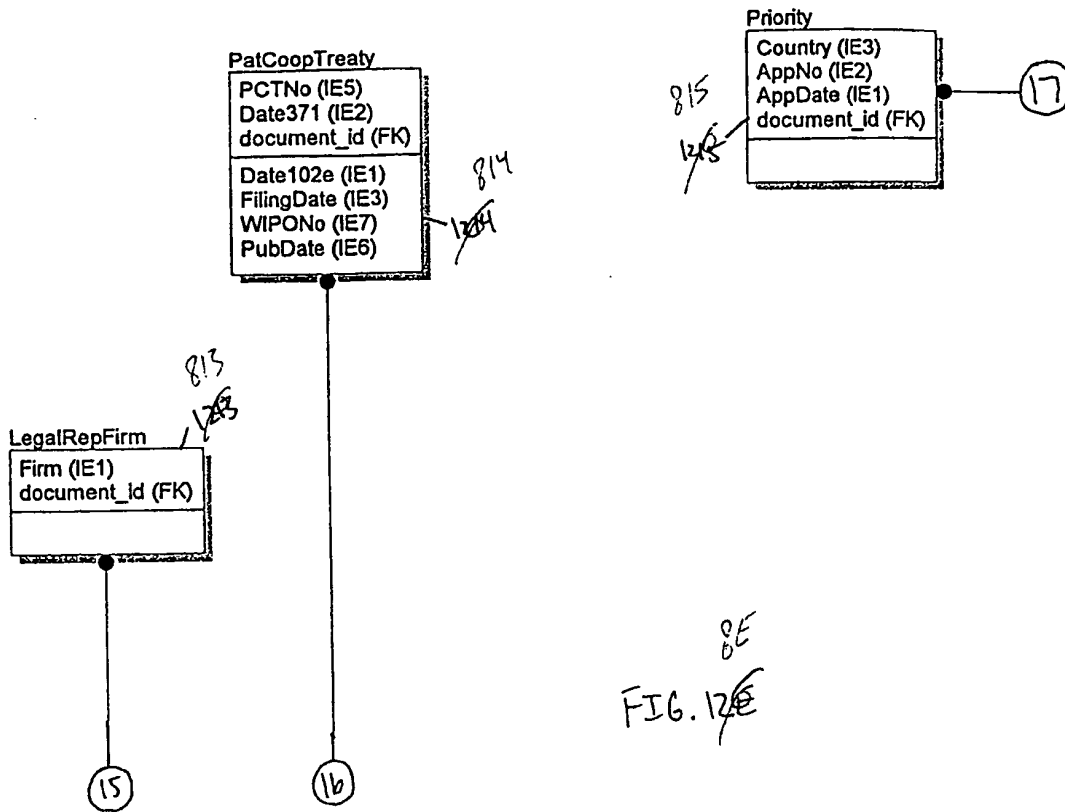
10 / 193



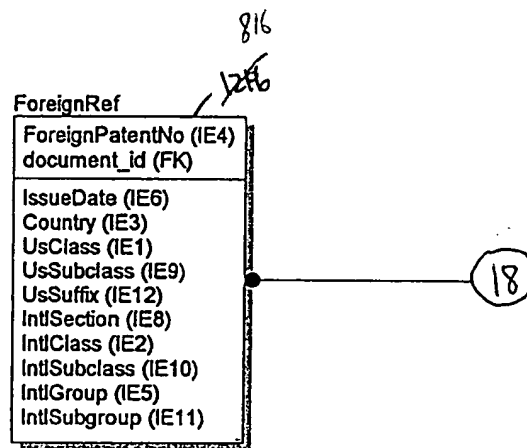
11/193

8D  
FIG. 126

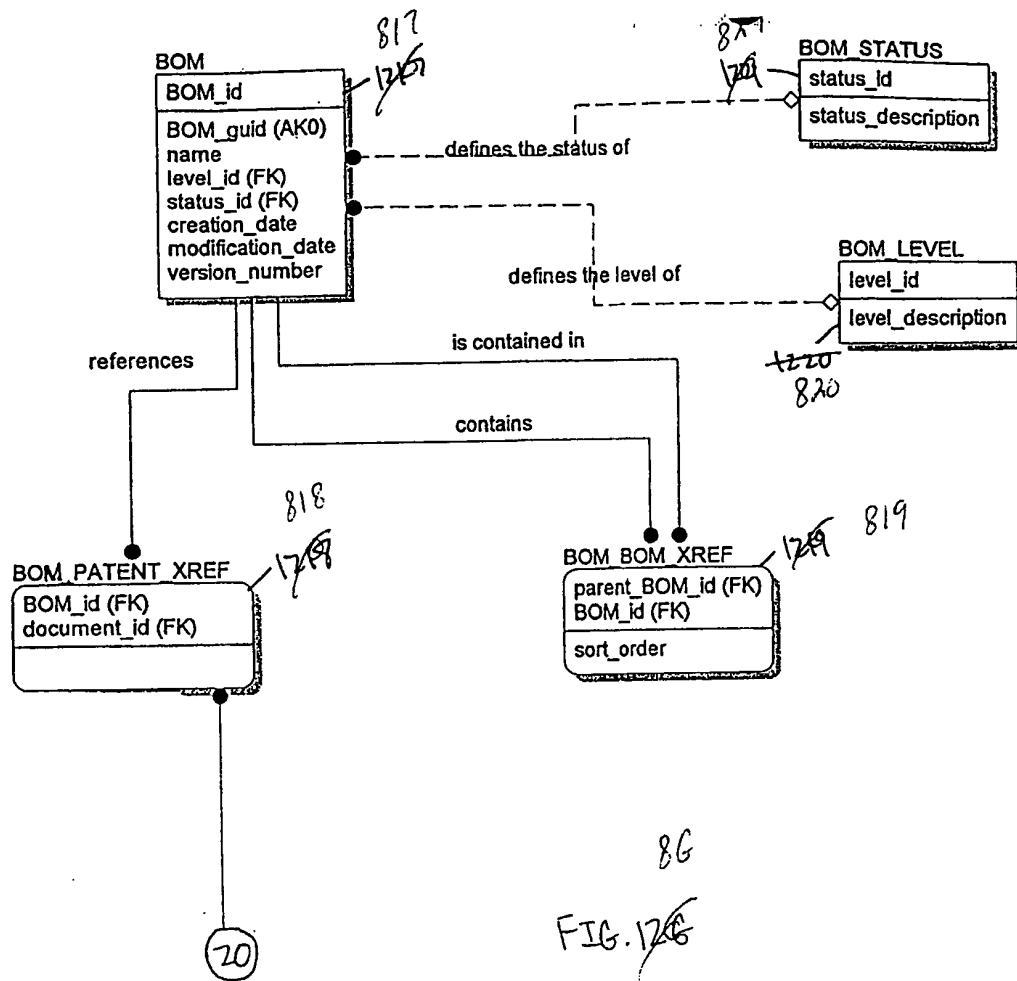
12/193



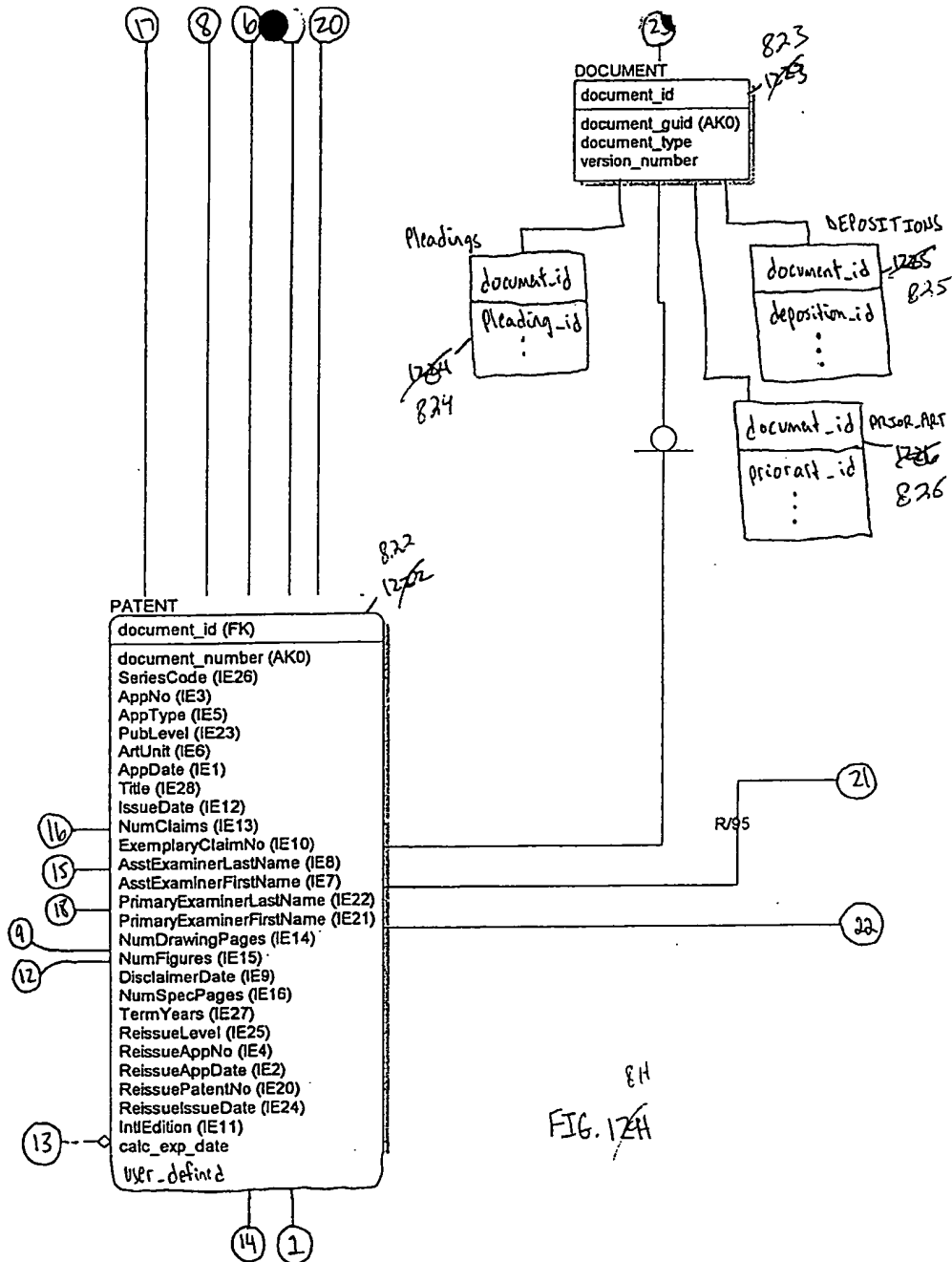
13/193

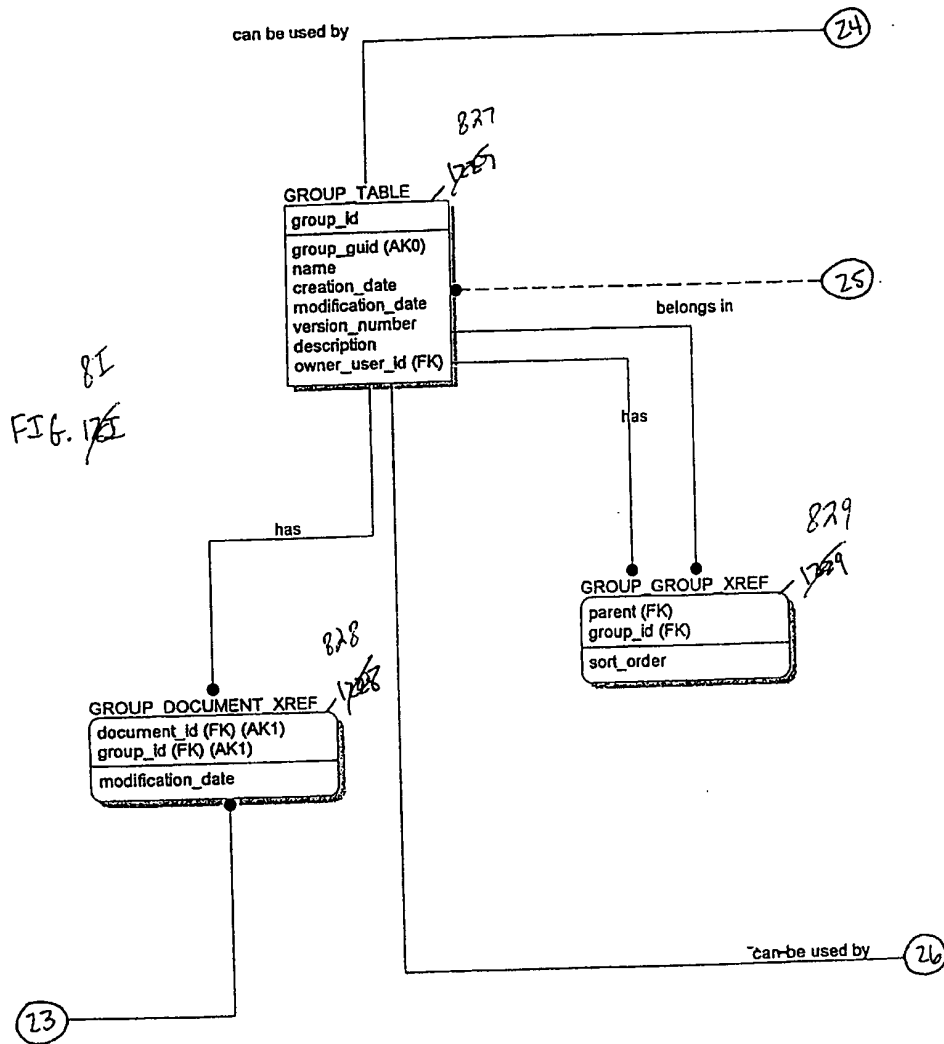


8F  
FIG. 124



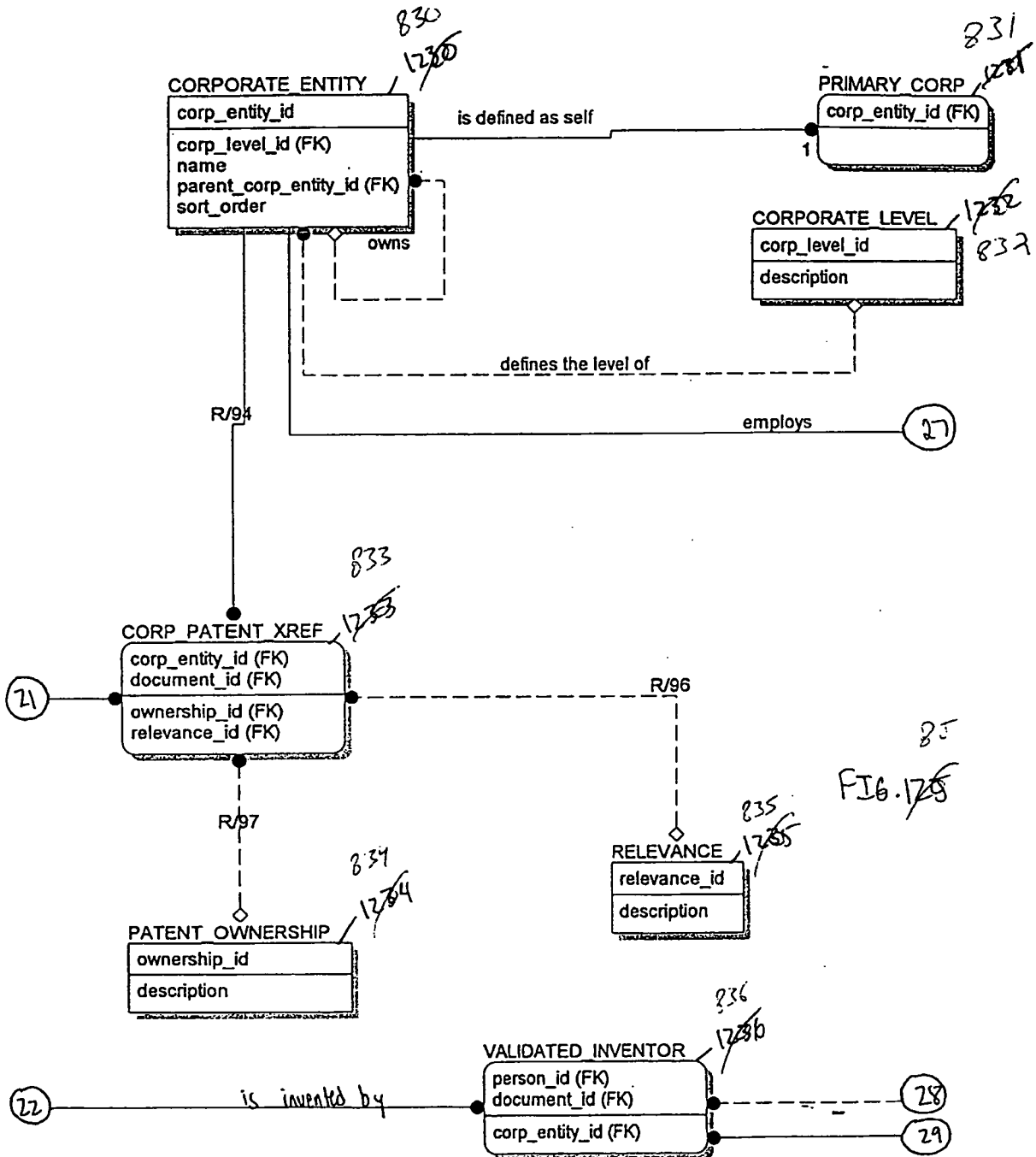
15/193



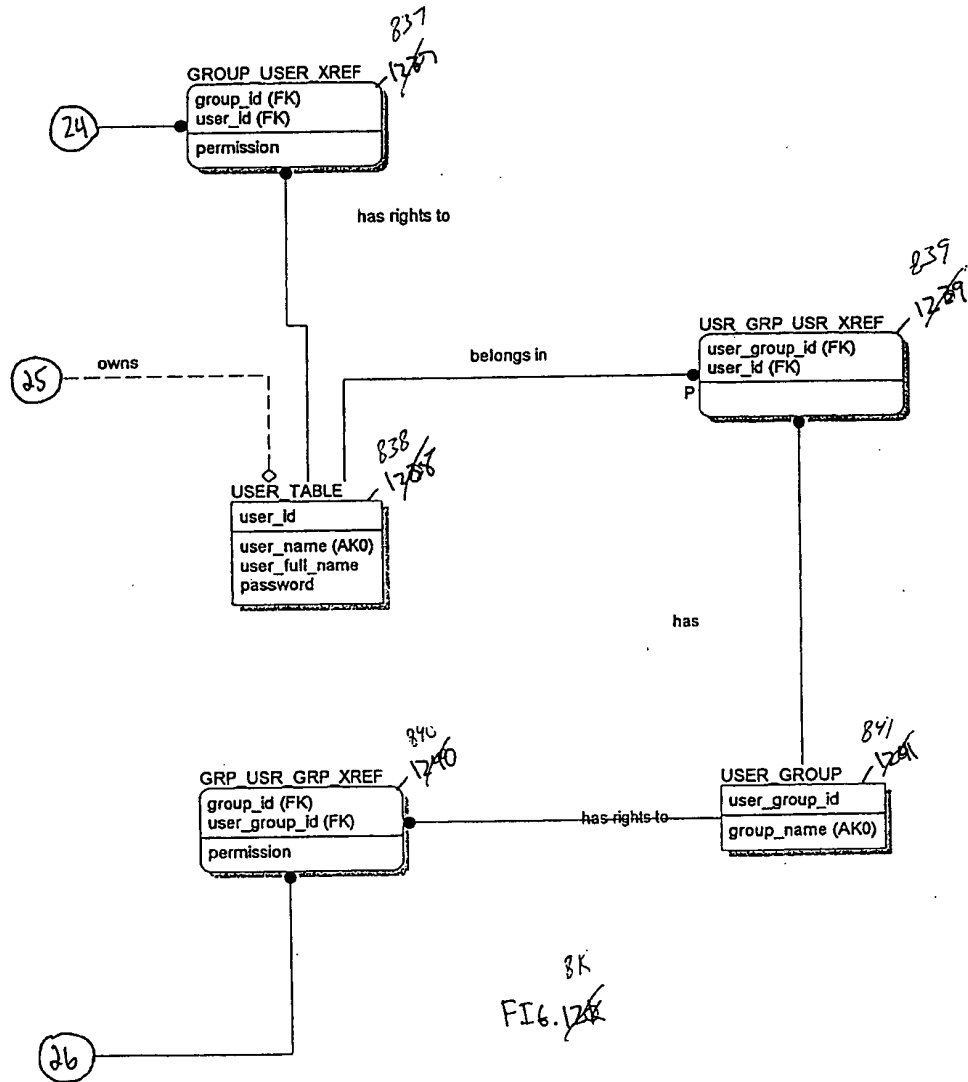




17/193



18/193



19/193

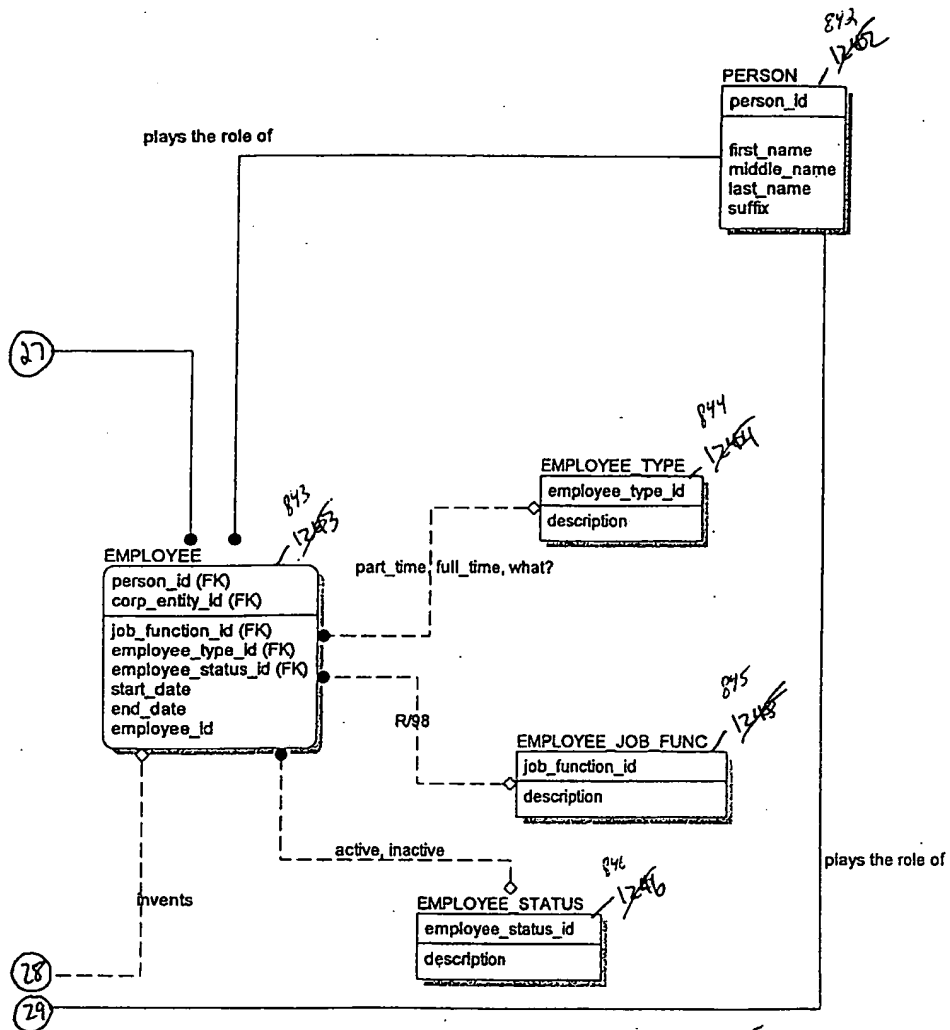


FIG. 102  
8L

20/193

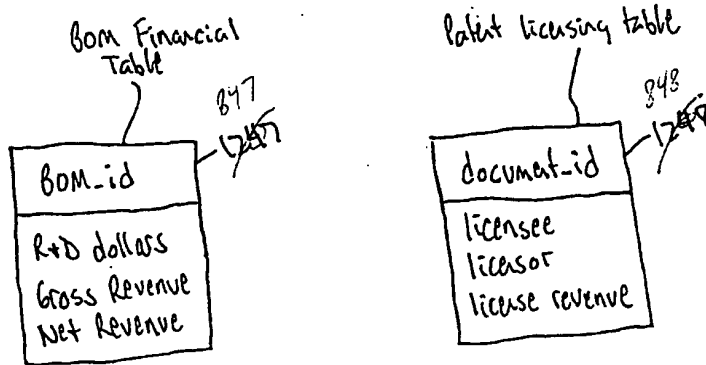
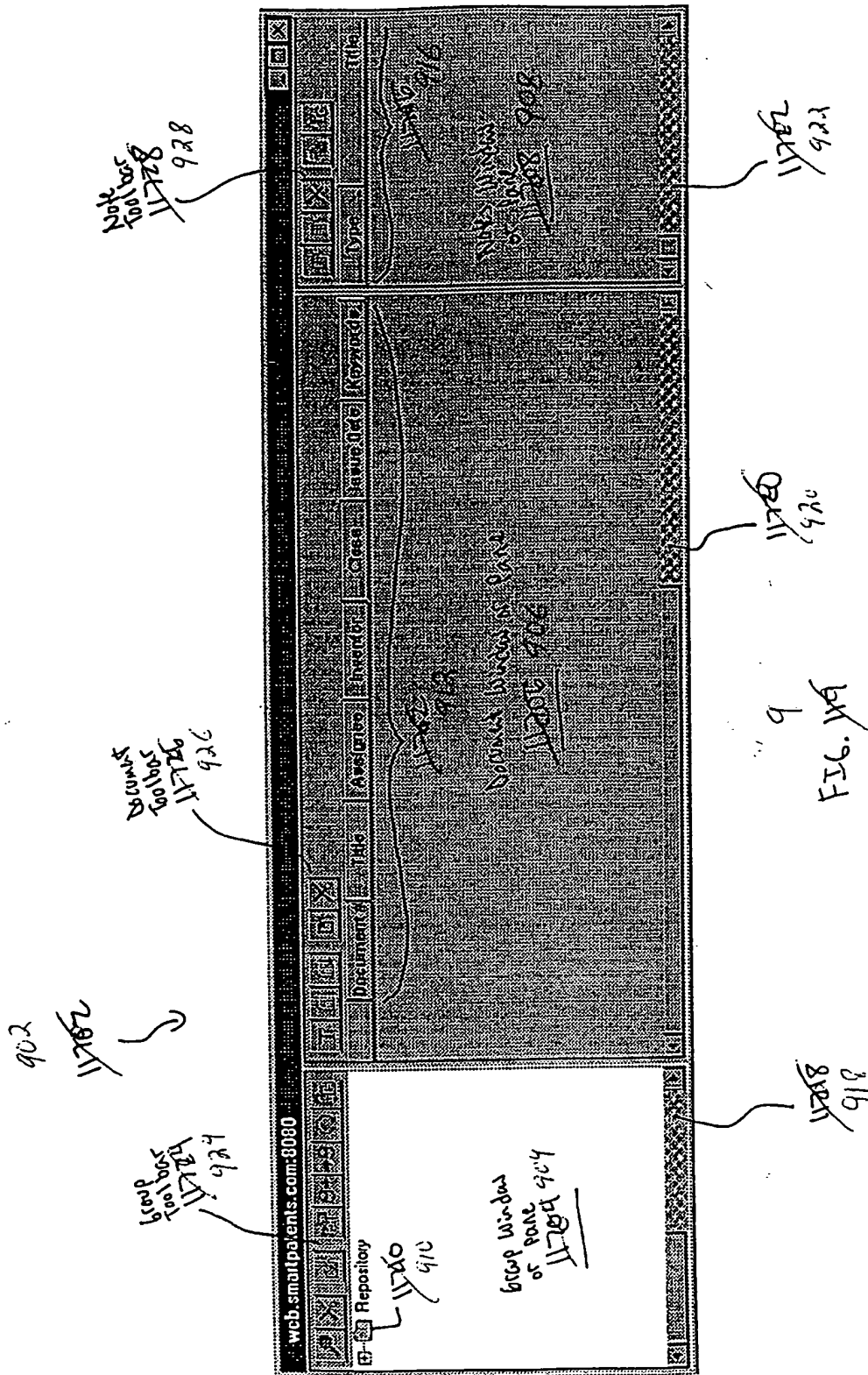
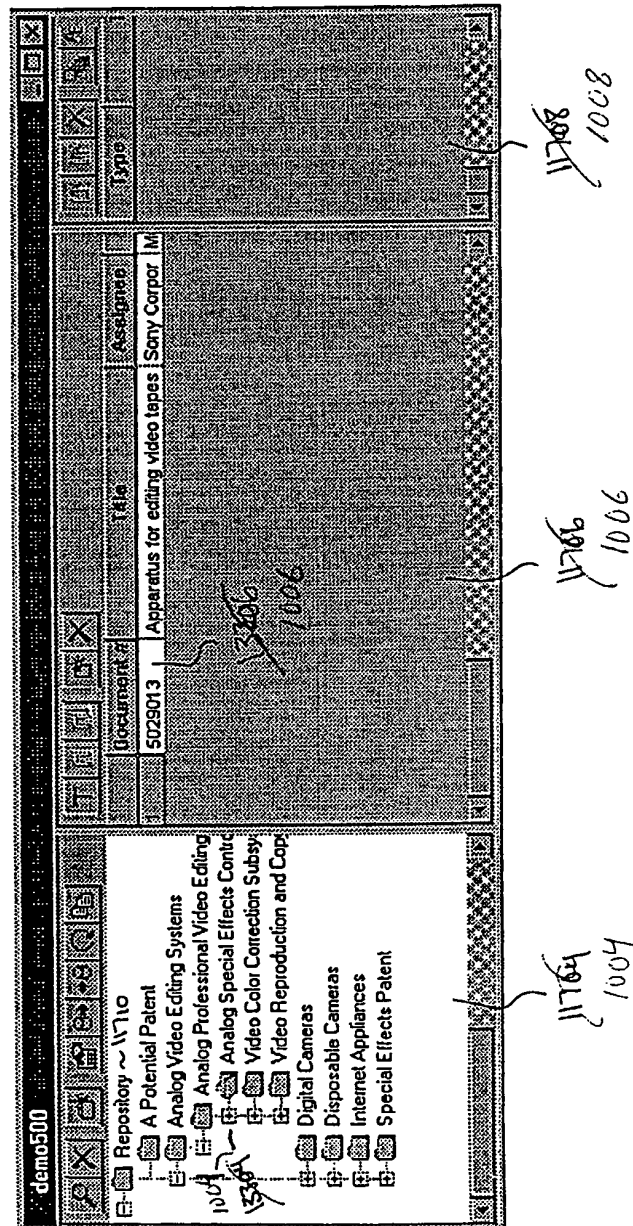


FIG. 12M



22/ 193



23/193

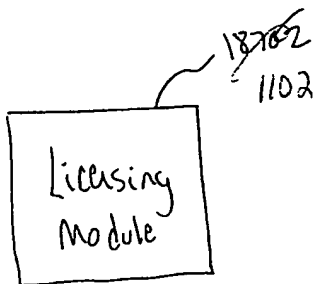


FIG. 18762  
11A

24/193

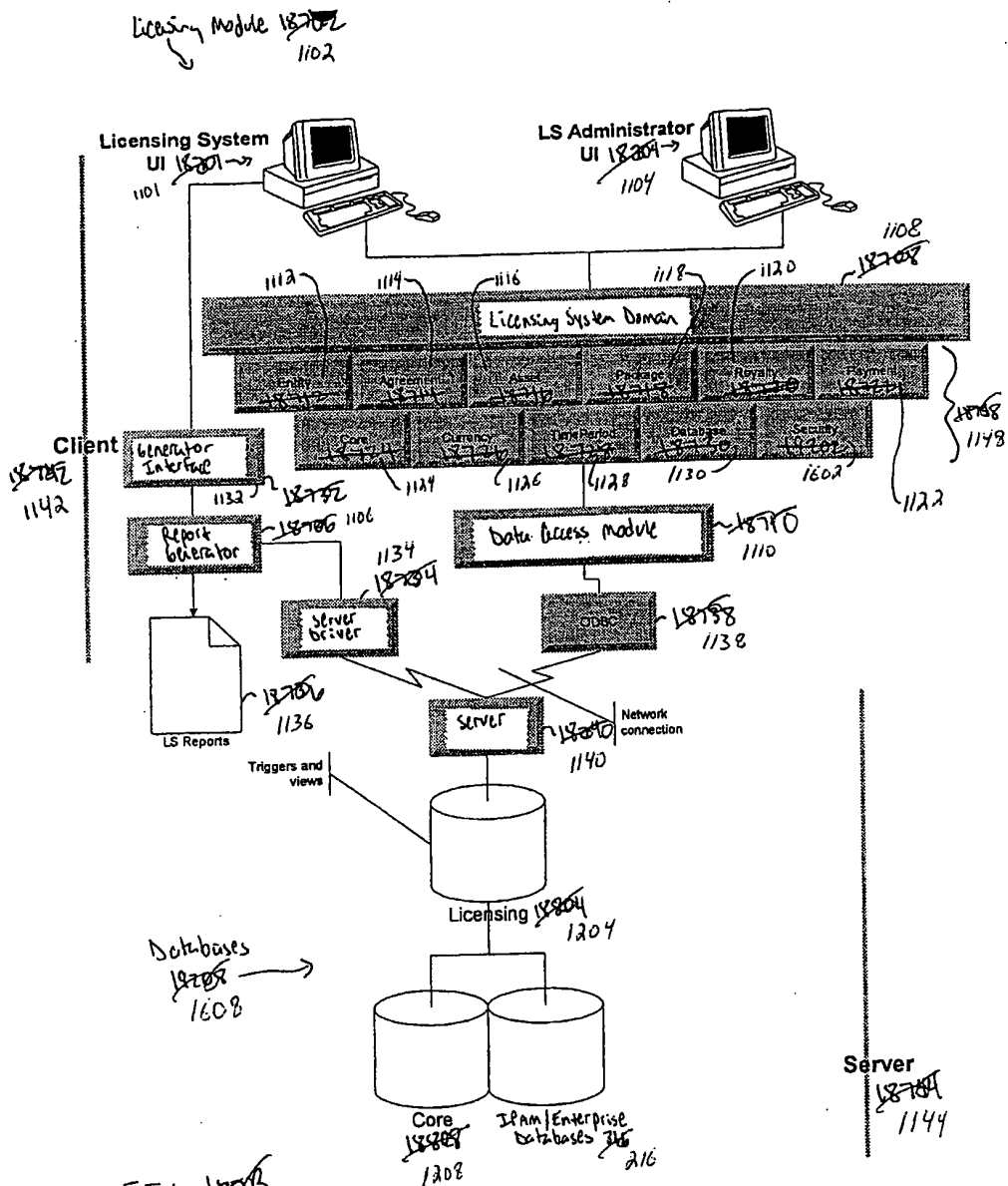


FIG. 11B



25/193

Standalone Configuration ~~1801~~ 1201 ↘

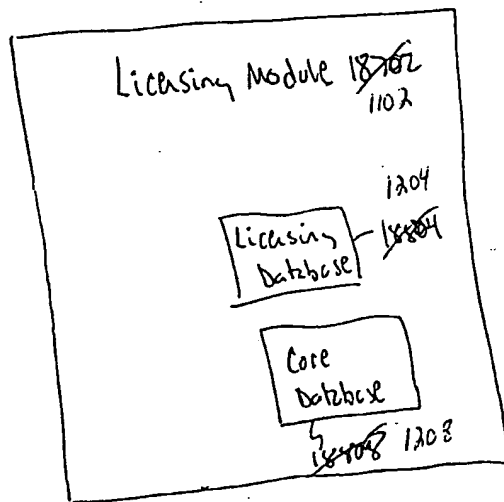


FIG. ~~1803~~ 12

26/193

Integrated Configuration 1302  
18902 ↓

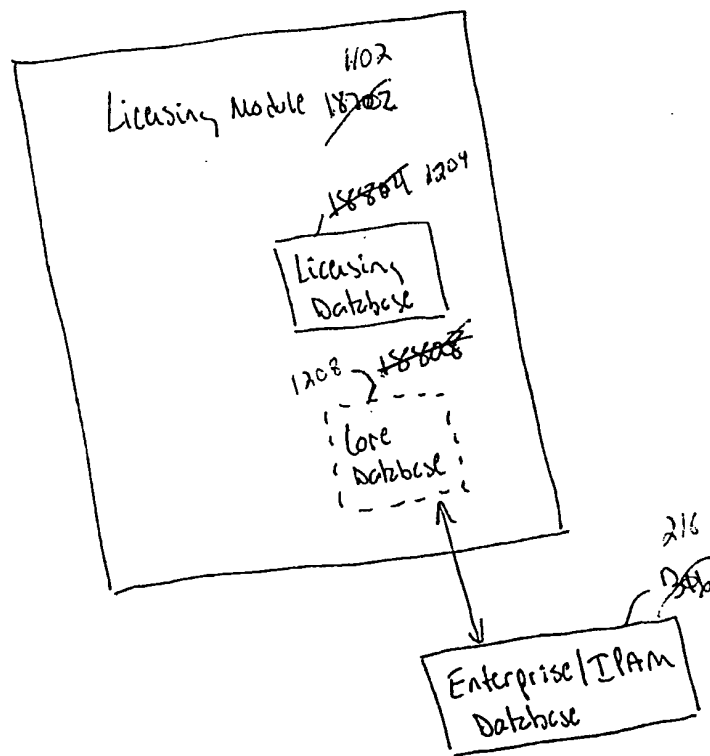
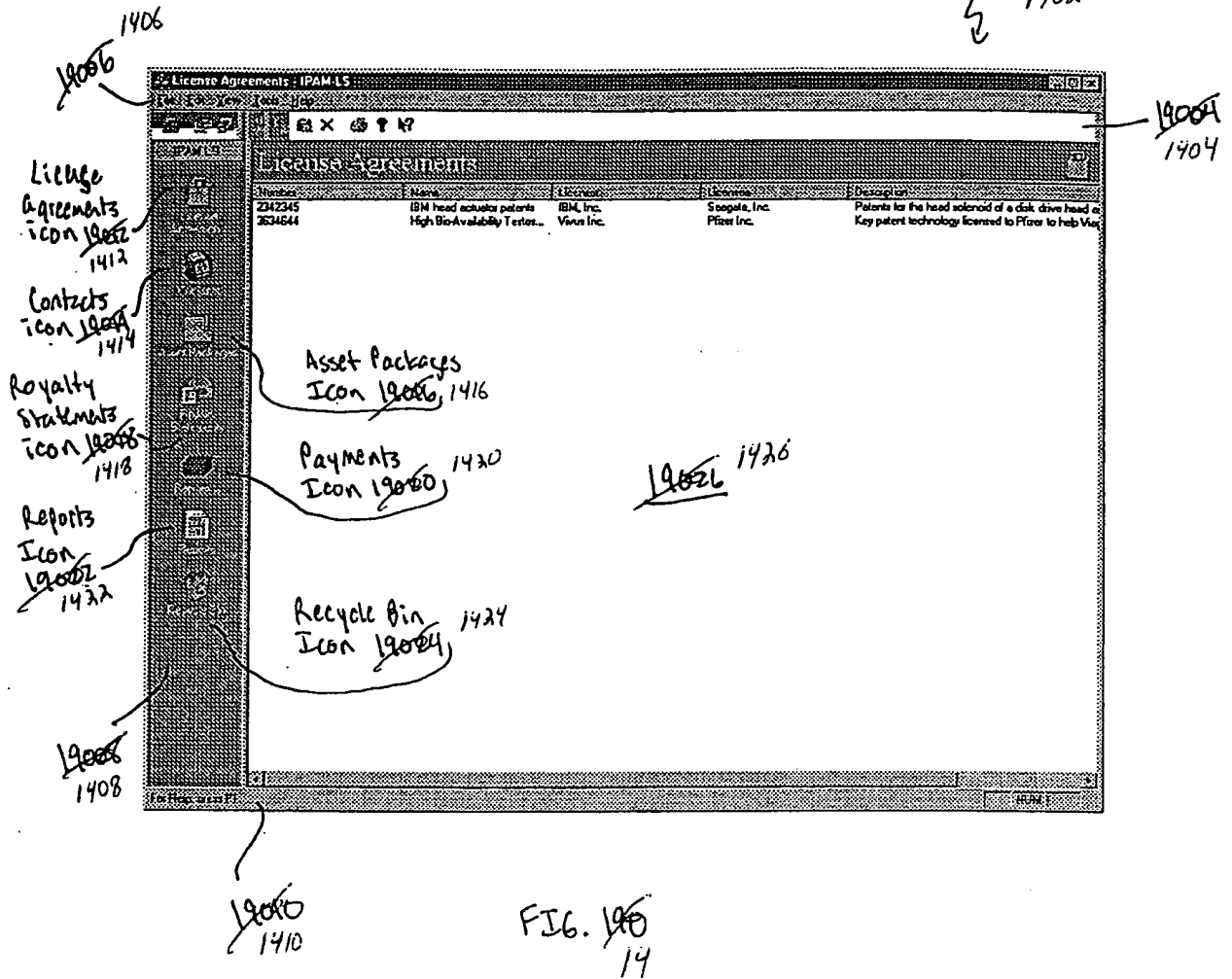


FIG. 13  
13

27/193

Object Display Window  
14002  
1402



28/193

## Actor Hierarchy 191027

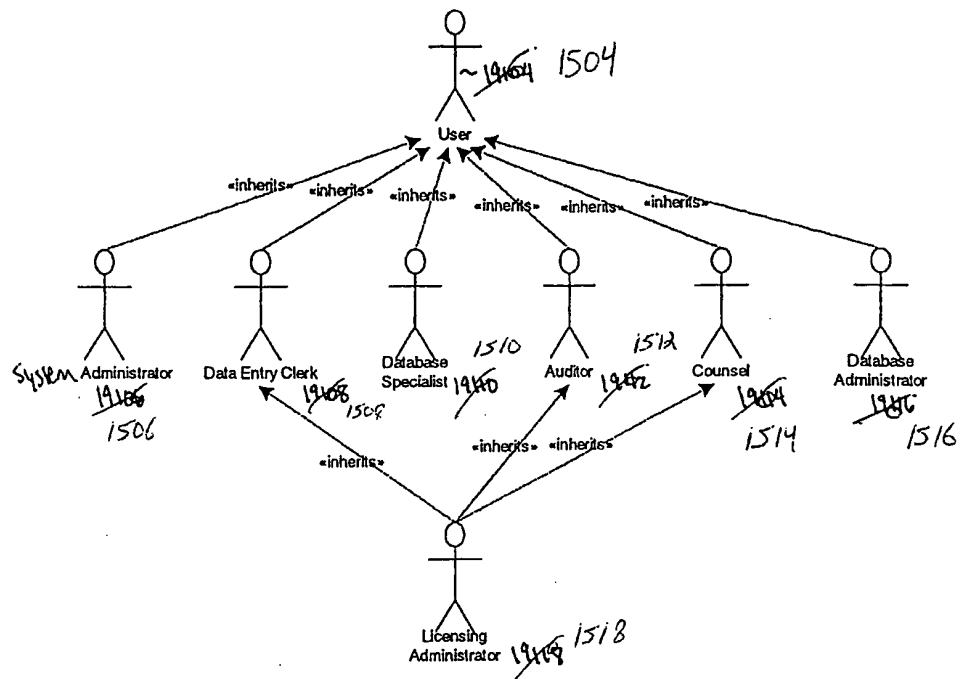


FIG. 15

29/193

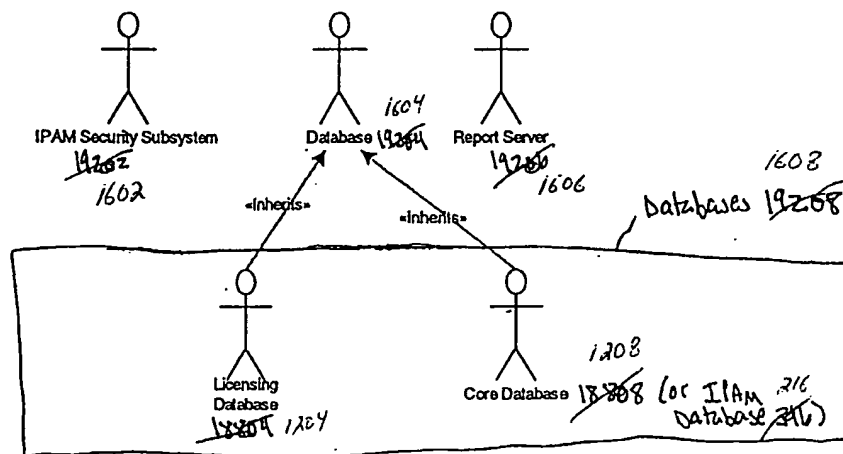


FIG. 19c  
16

30/193

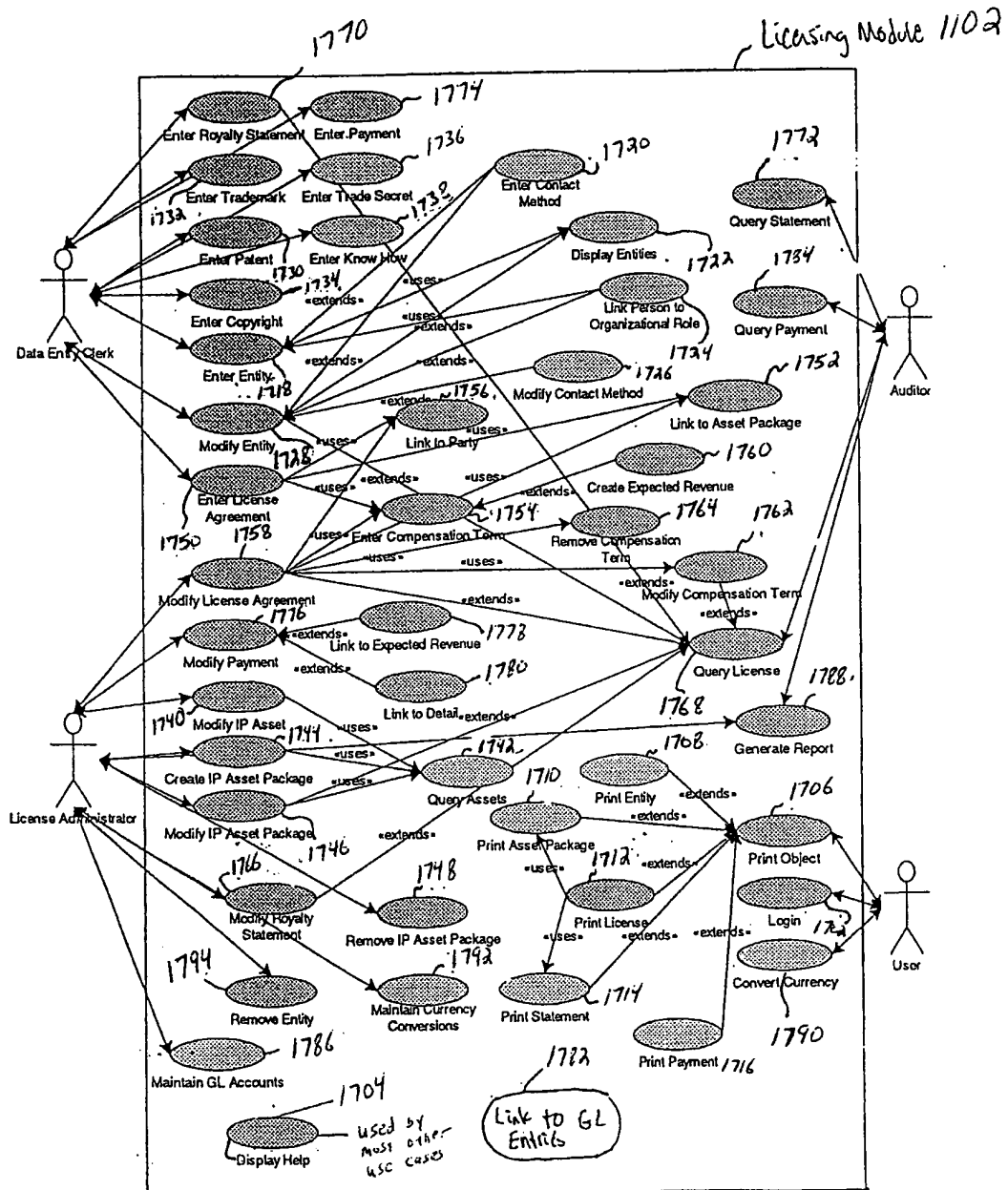
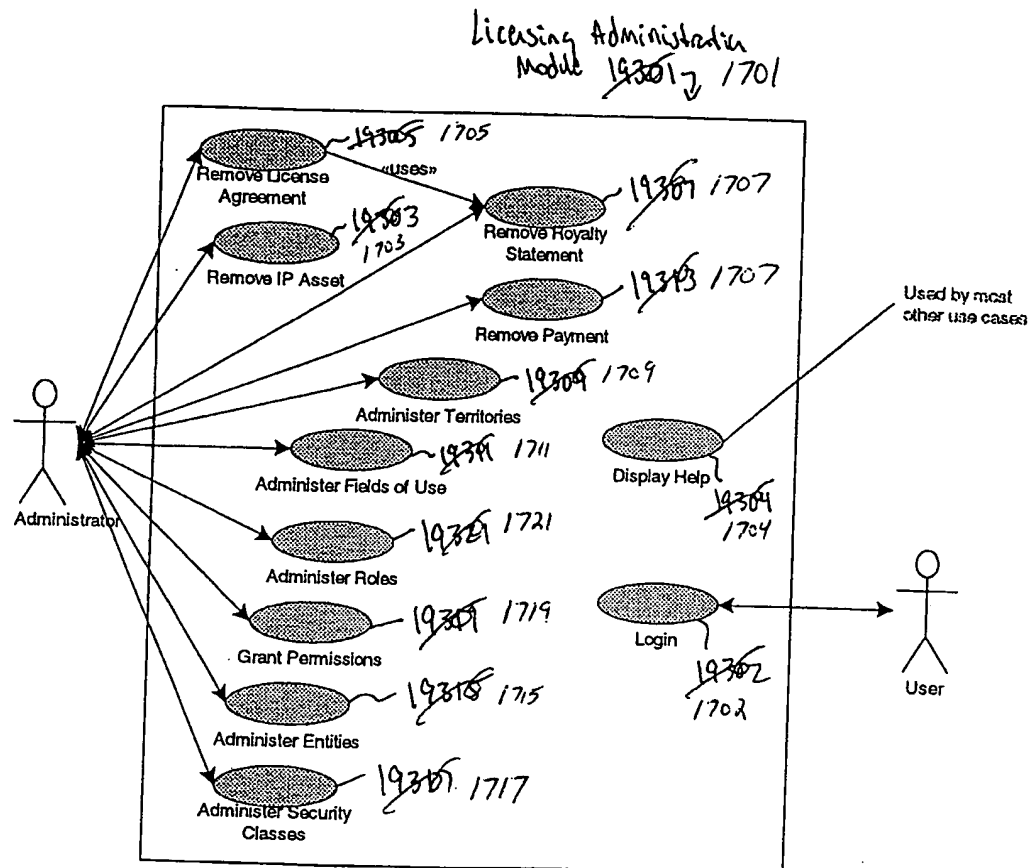
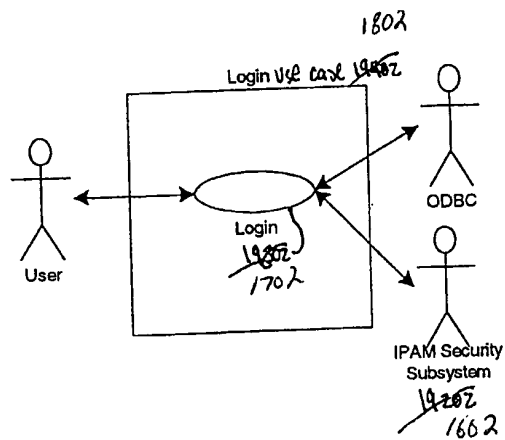


FIG. 17A

31/193

17B  
FIG. 19308

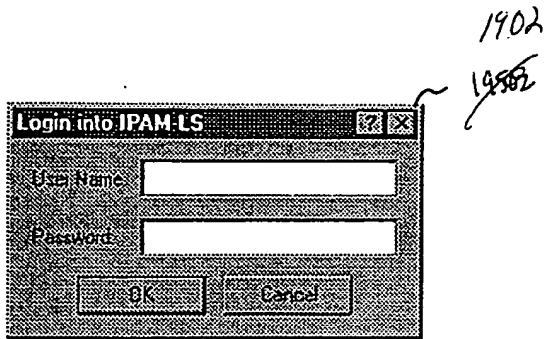
32/193



18  
FIG. 10A

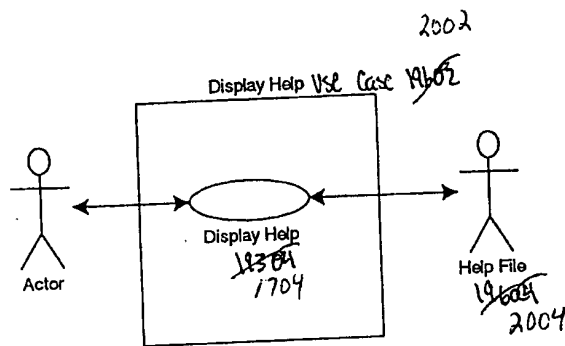


33/193



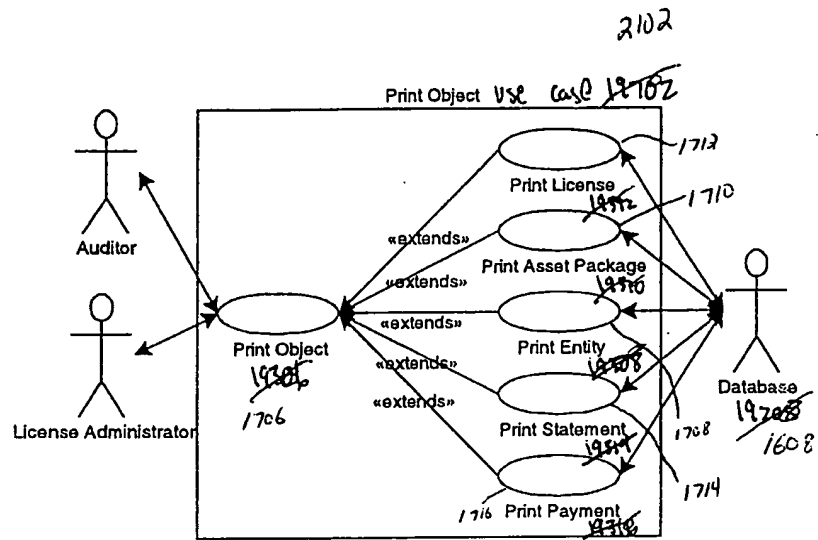
19  
FIG. 19

34/193



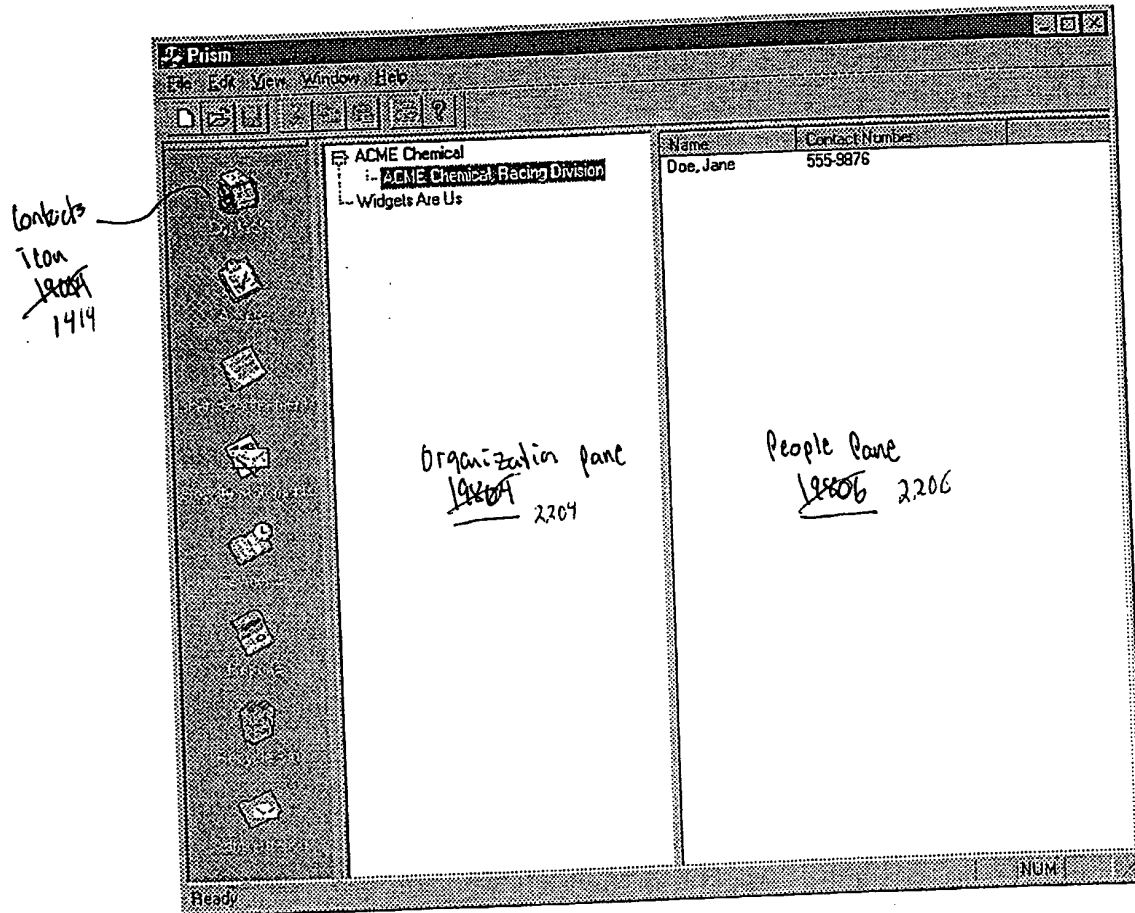
20  
FIG. 196

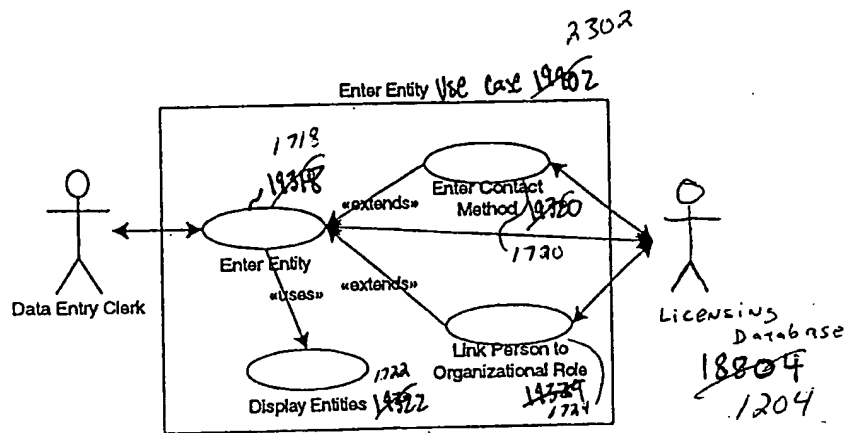
35/193



21  
FIG. 196

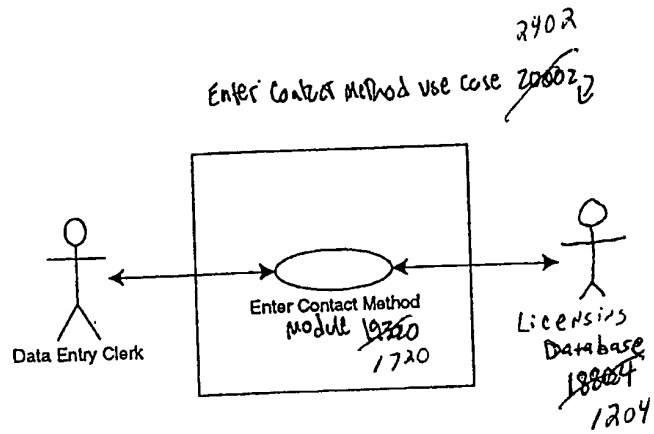
36/193

Contact View <sup>2202</sup>  
~~19802~~22  
FIG. 19E



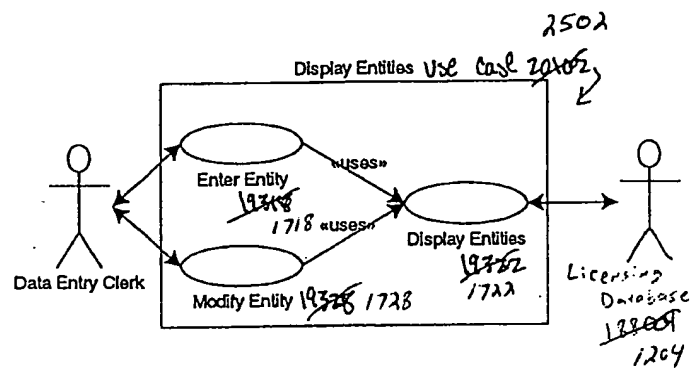
23  
FIG. 194

38/193



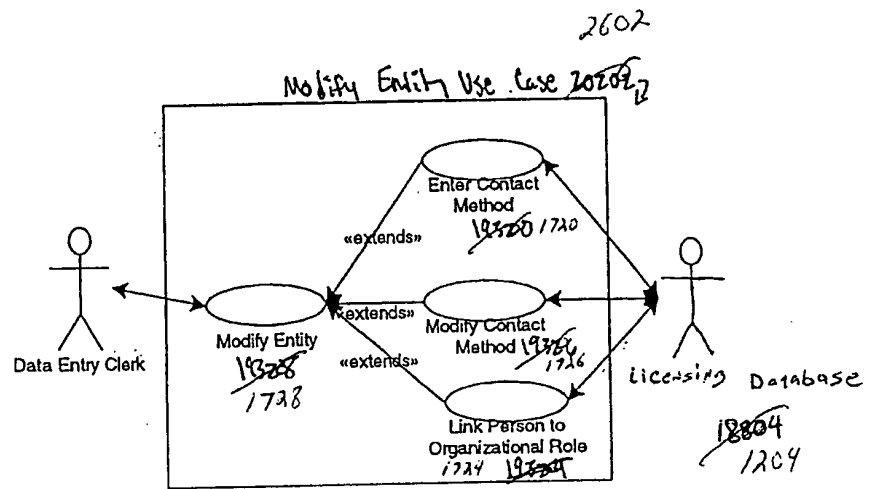
24  
FIG. 200

39/193



25  
FIG. 25

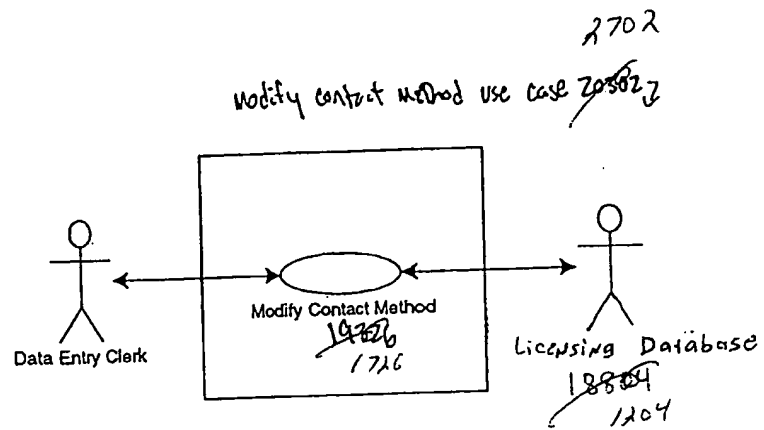
40/193



26  
FIG. 202

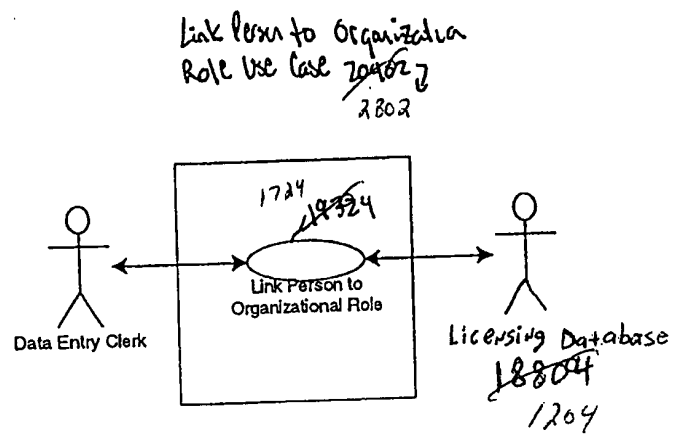


41/193



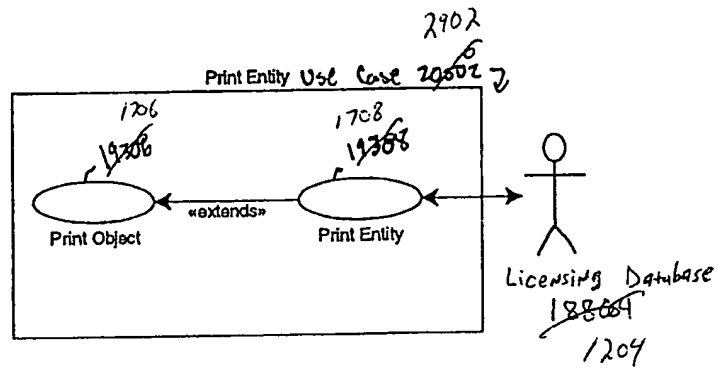
27  
FIG. 203

42/193



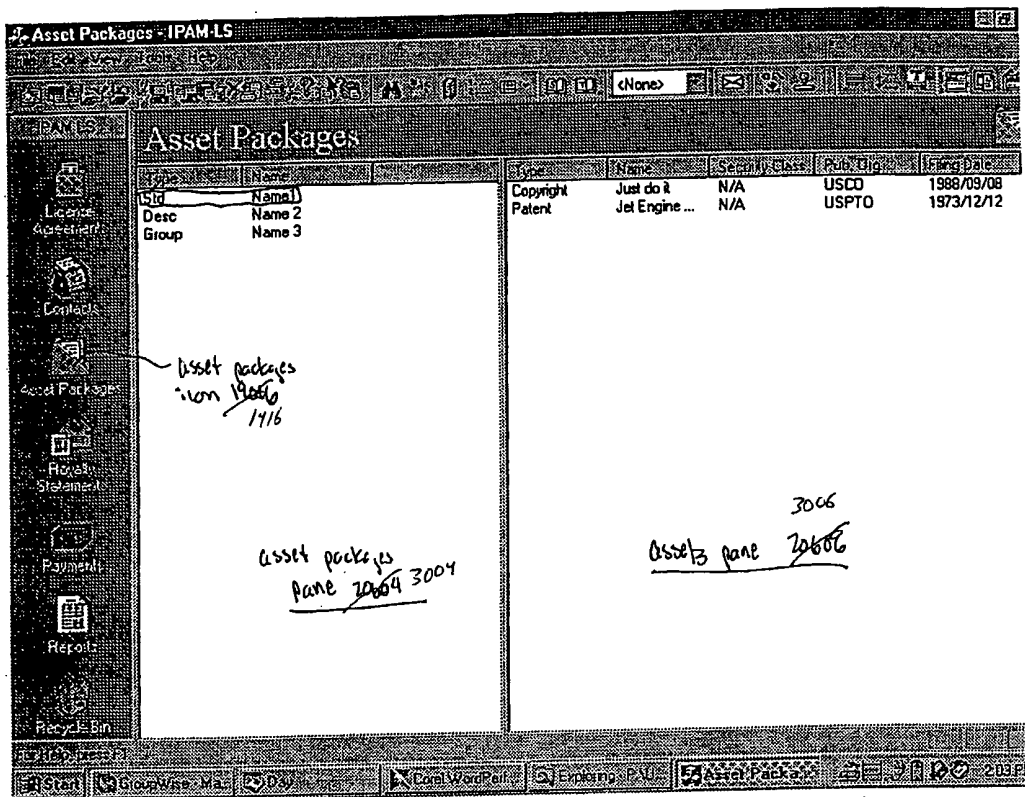
28  
FIG. ~~204~~

43/193

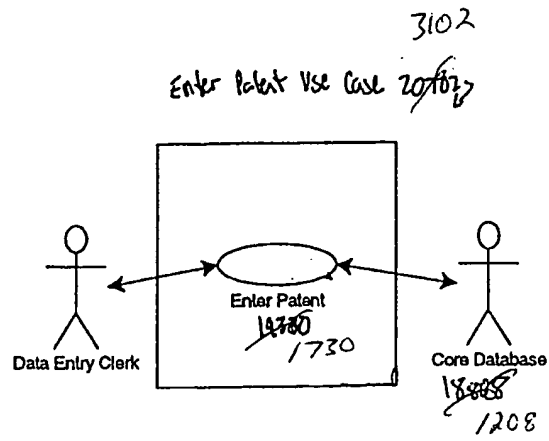


29  
FIG. 29

44/193

3002  
Asset view 2060230  
Fig. 206

45/193

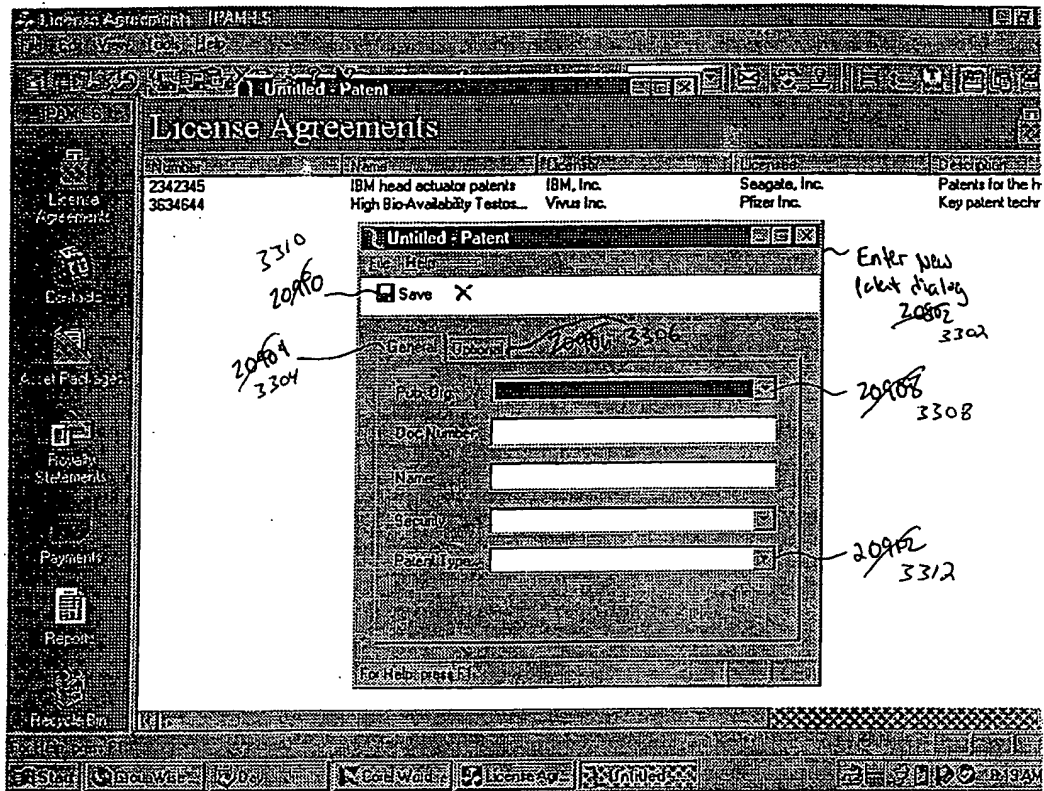


31

Fig. 207



47/193



33  
FIG. 209

48/193

License Agreements: IP2M-1.5

License Agreements

Number	Name	Licensee	Licensee	Description
2342345	IBM head actuator patents	IBM, Inc.	Seagate, Inc.	Patents for the h
3634644	High Bio-Availability Testos...	Vivus Inc.	Pfizer Inc.	Key patent techn

Untitled - Patent

Save X

Pat No.

Pat Number

USPTO  
EPO  
USCO  
JTO

Name

Country

Patent type

20982 3302

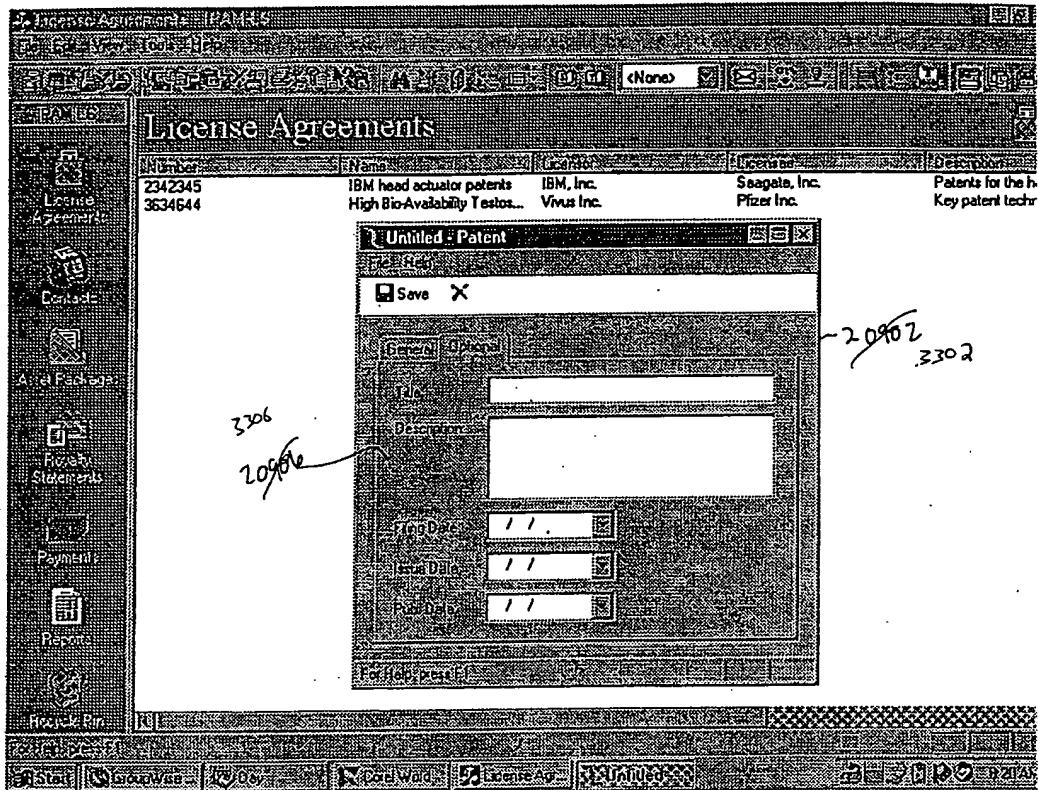
21002 3402

34

FIG. 2/0

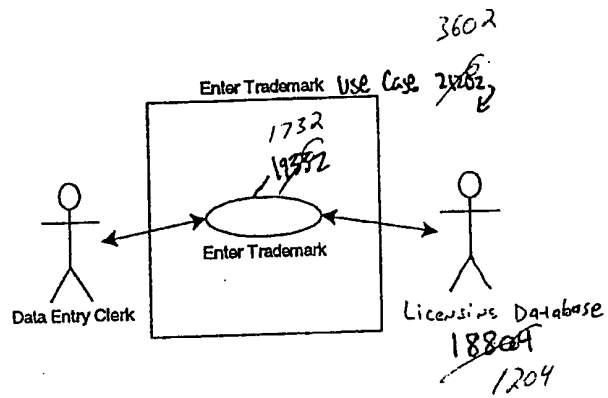


49/193



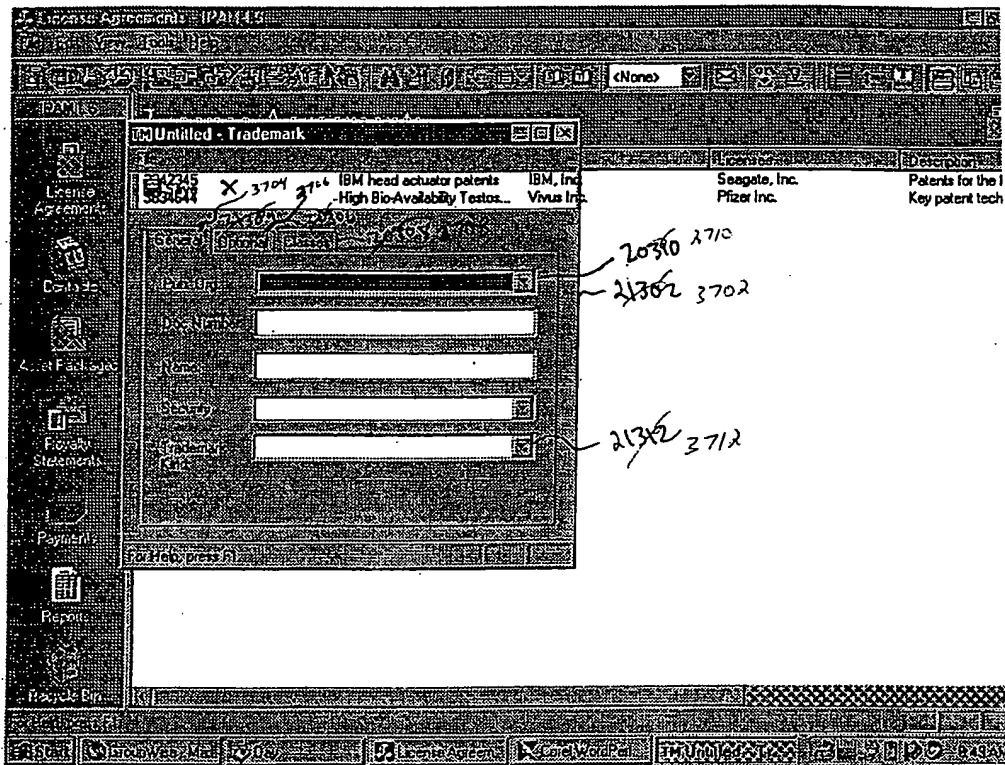
35  
FIG. 2A

50/193



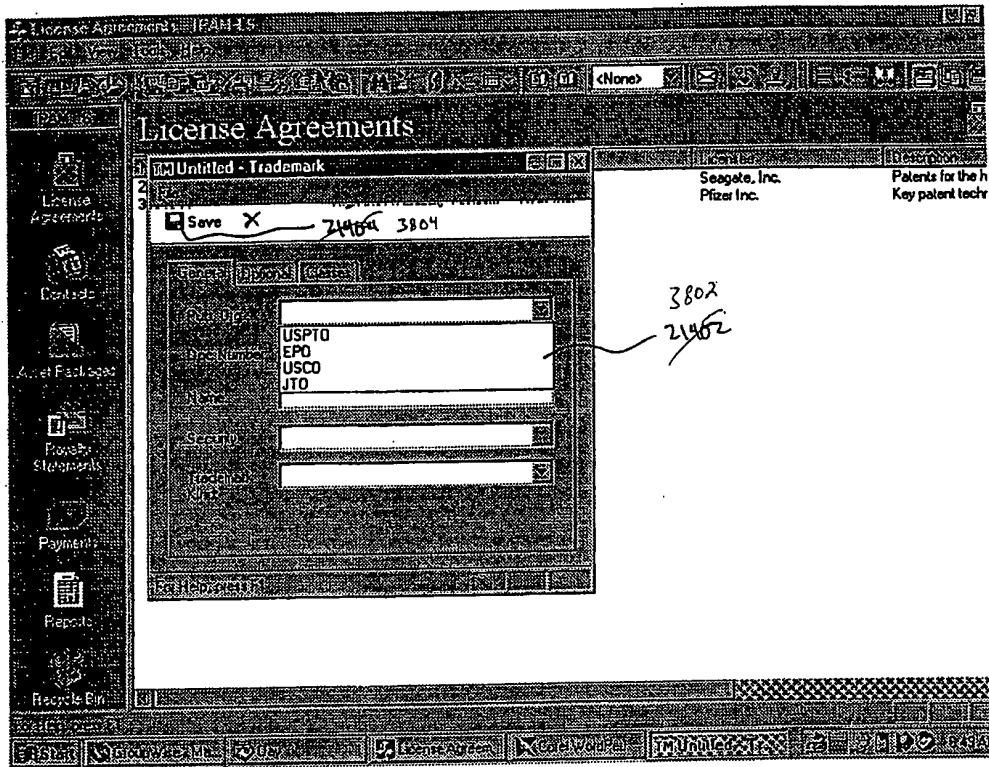
36  
FIG. 246

51/193



37  
FIG. 248

52/193



38  
FIG. 2A

53/193

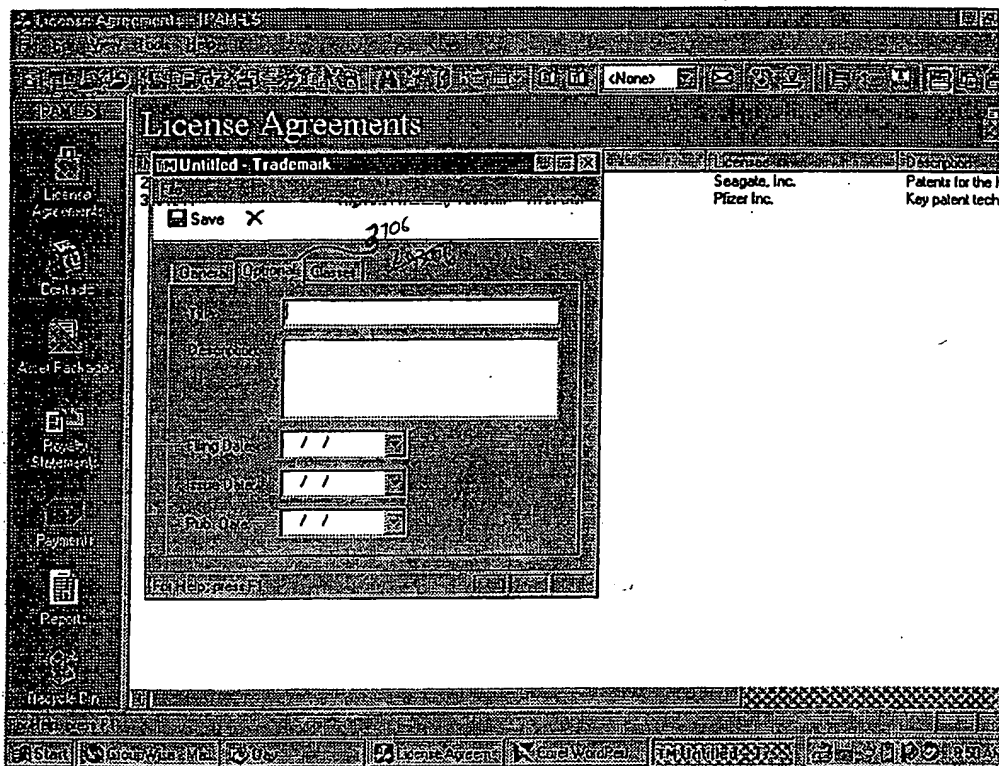
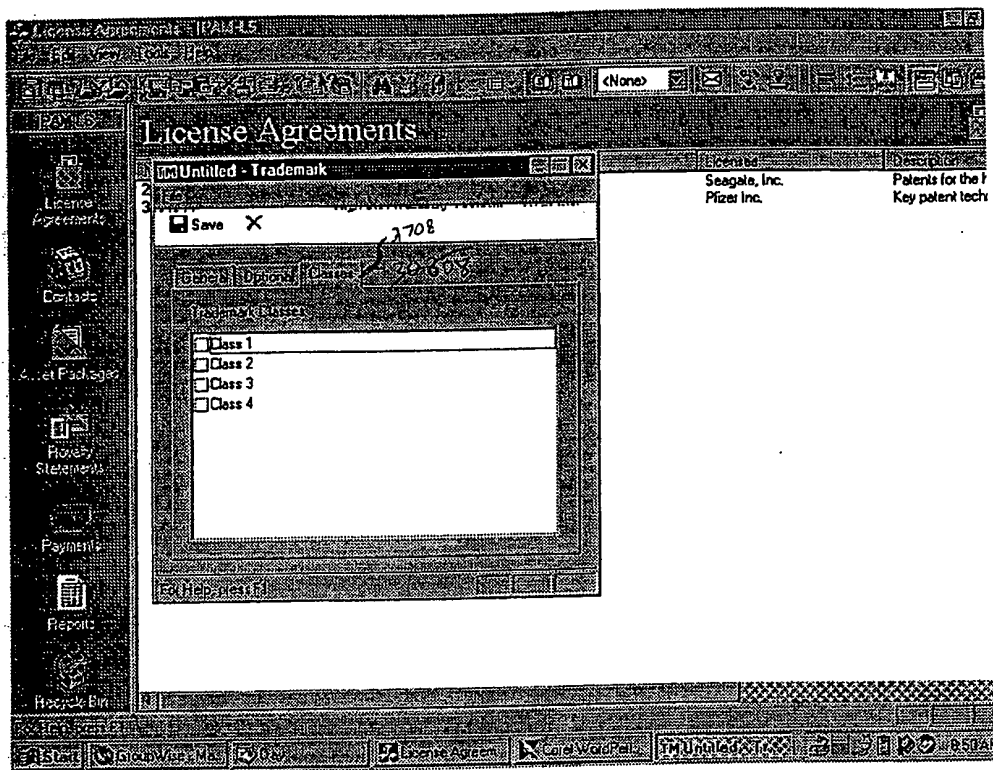
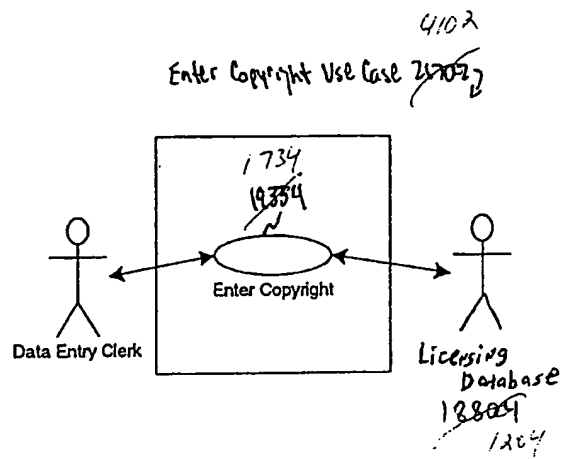


FIG. 218 39

54/193

FIG. 24<sup>40</sup>

55/193



41

Fig. 26

56/193

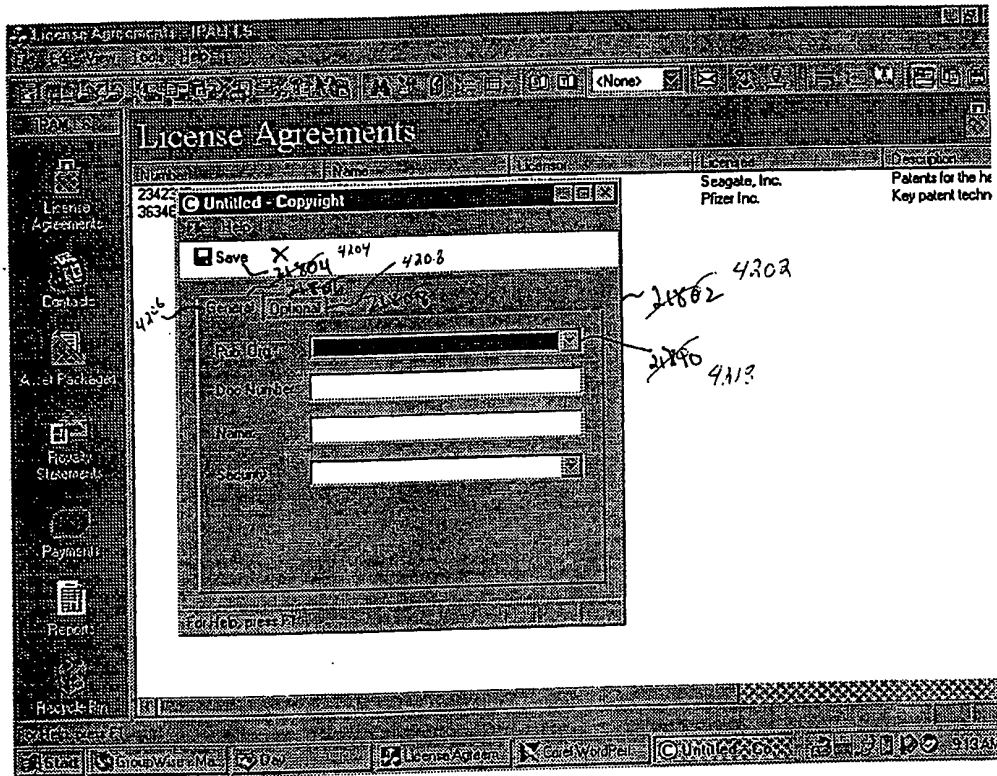


FIG. 2x8



5.7/193

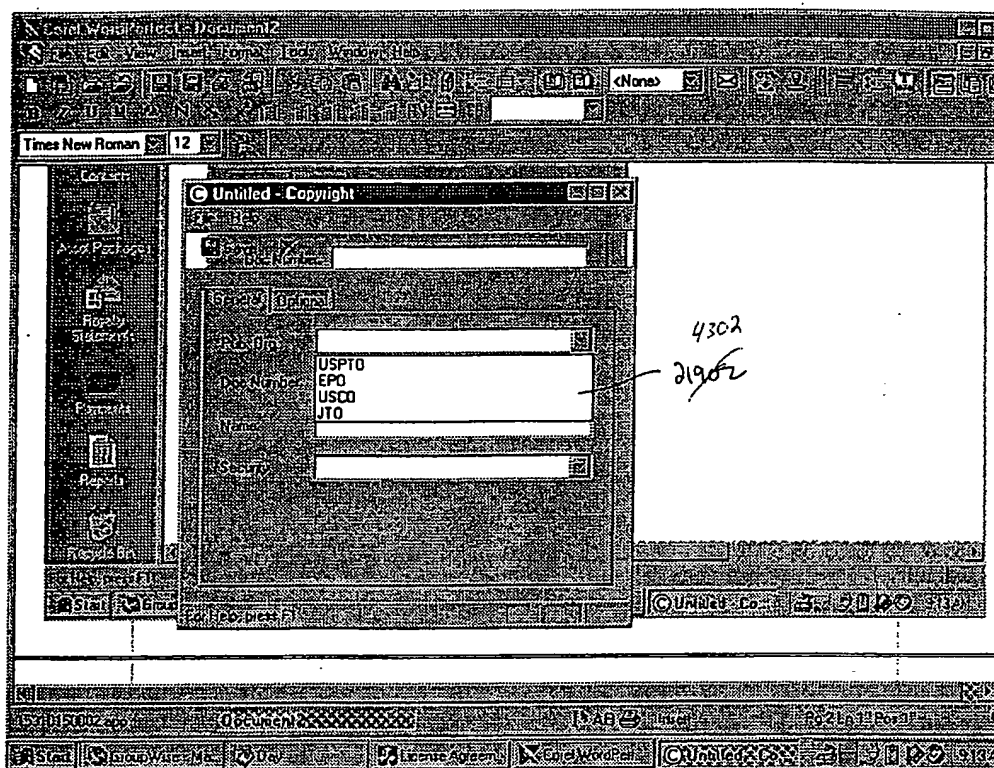
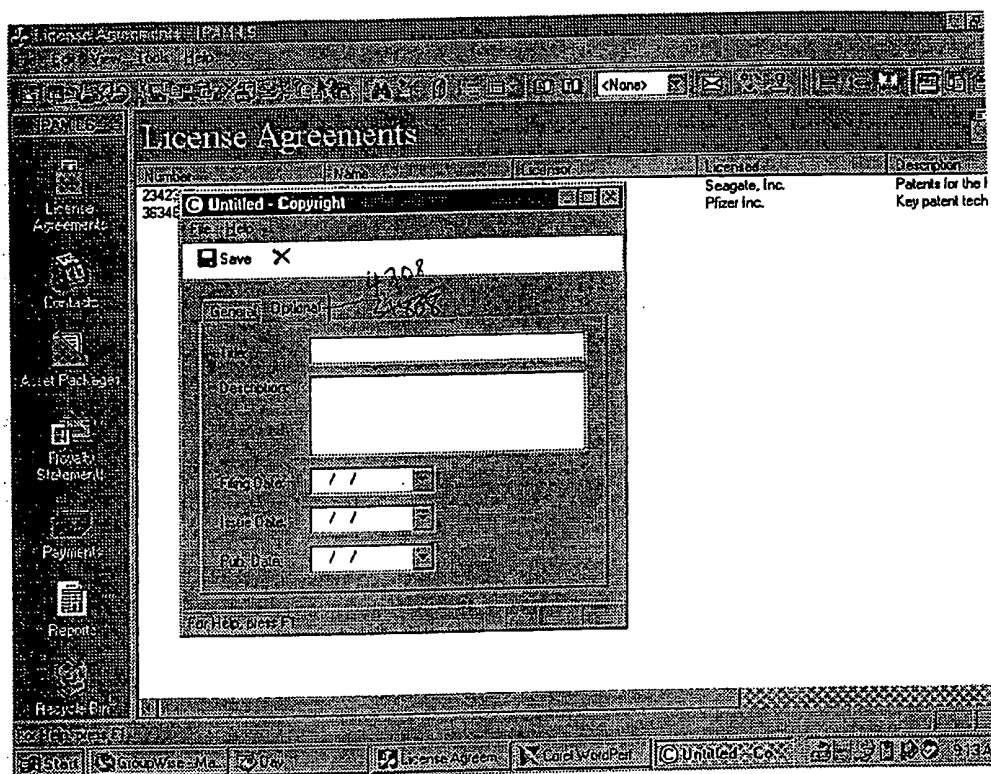


FIG. 219 <sup>43</sup>

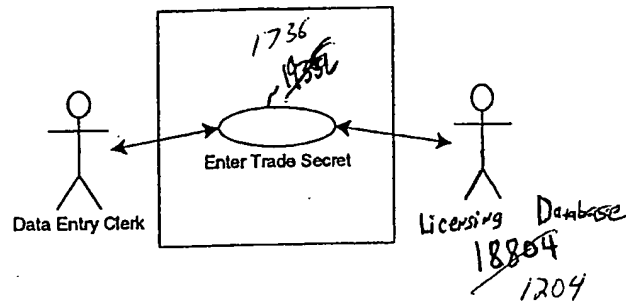
58/193

44  
FIG. 740

59/193

4502

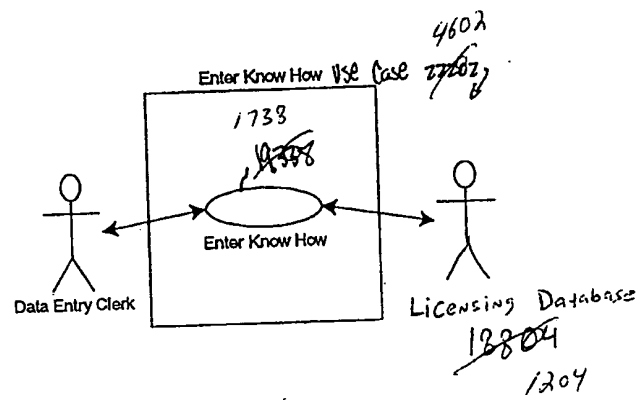
Enter Trade Secret use case ~~221822~~



45

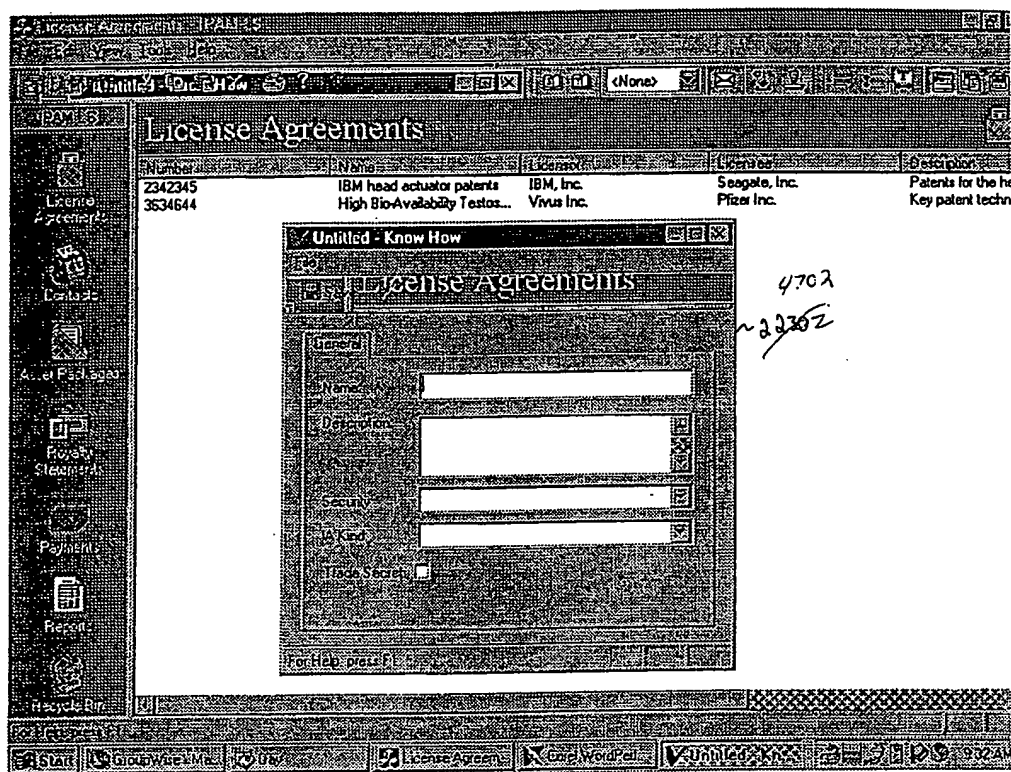
FIG. 2

60/193

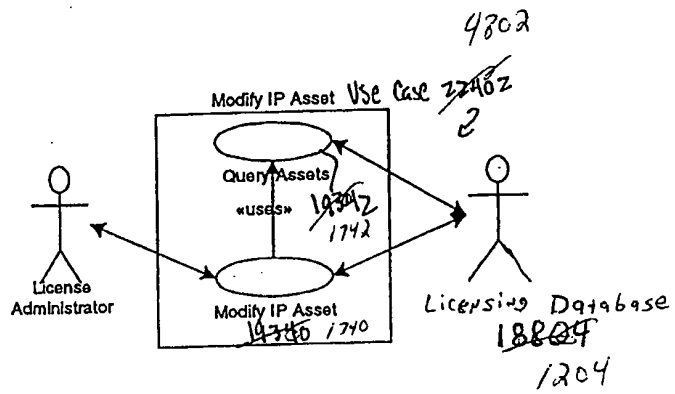


46  
FIG. 2762

61/193

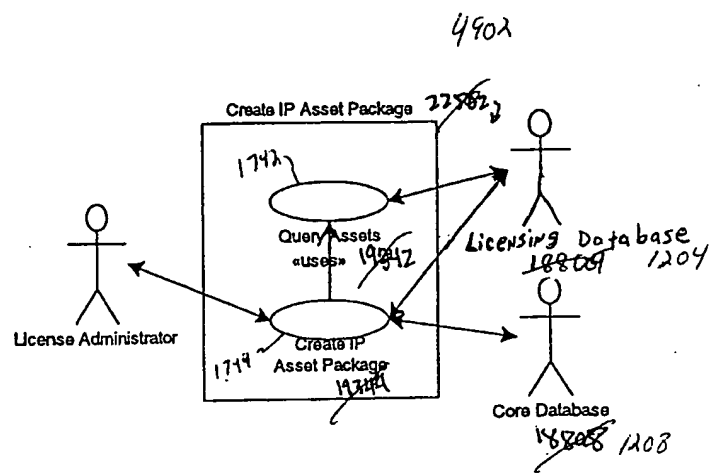
47  
FIG. 2/3

62/193

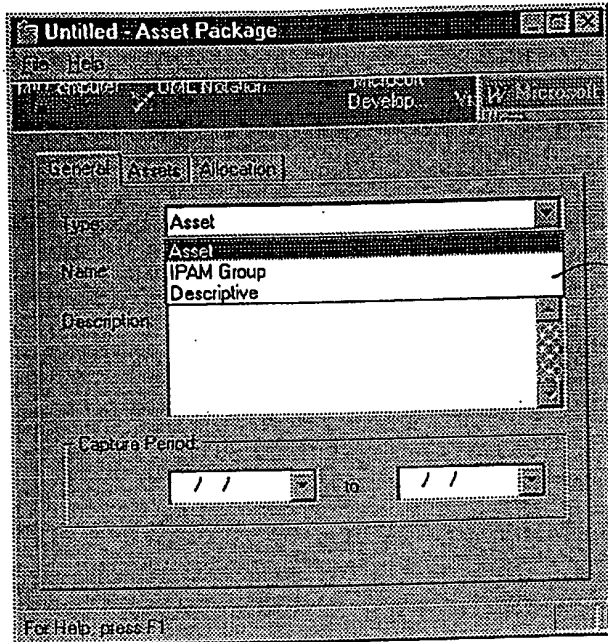


48  
FIG. 224

63/193



49  
FIG. 225



Asset Package Dialog 22602  
5002

22604 5004

FIG. 226 50



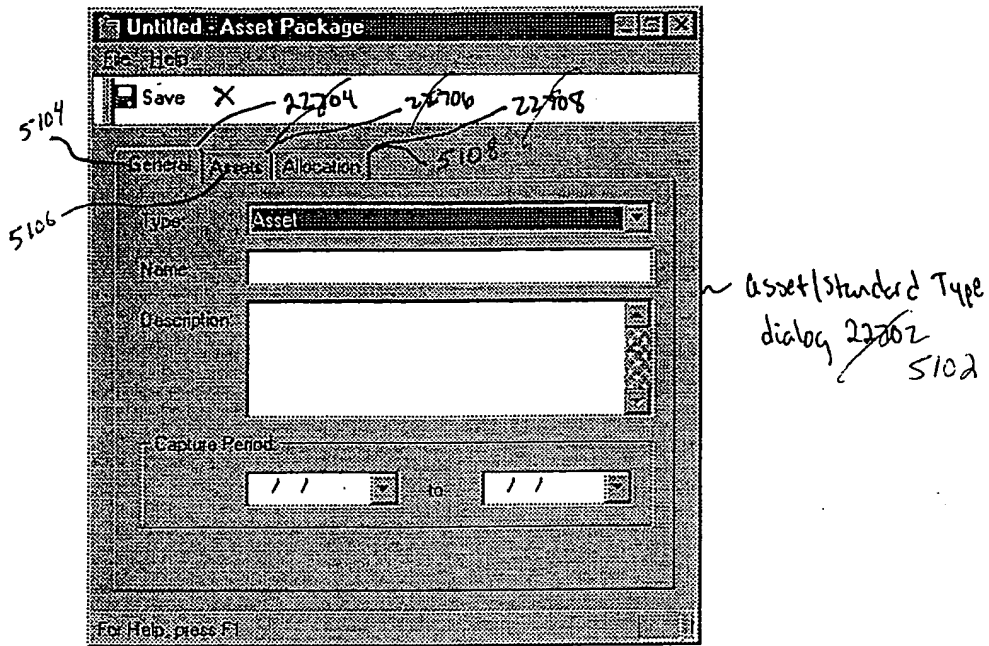
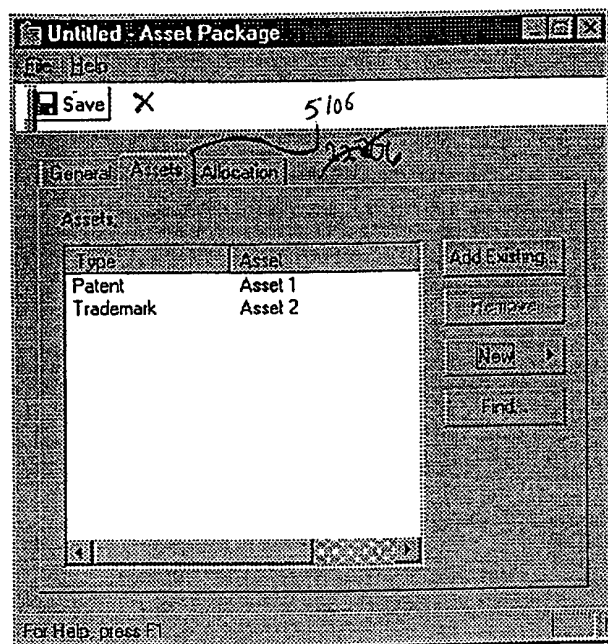
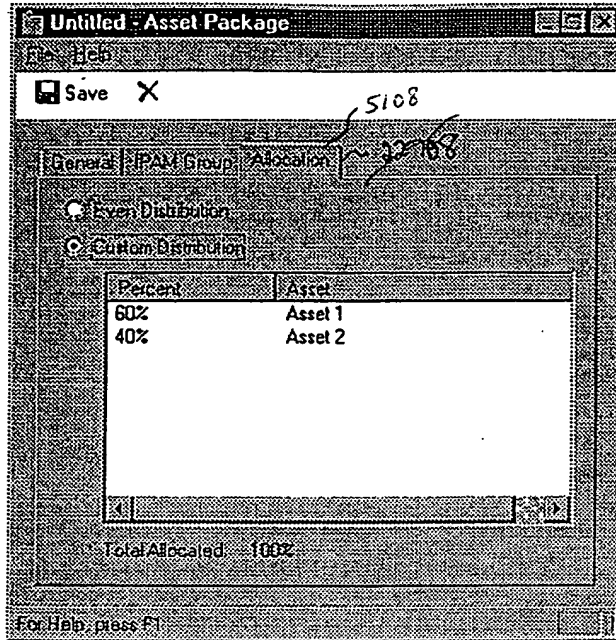


FIG. 267

66/193

52  
FIG. 228



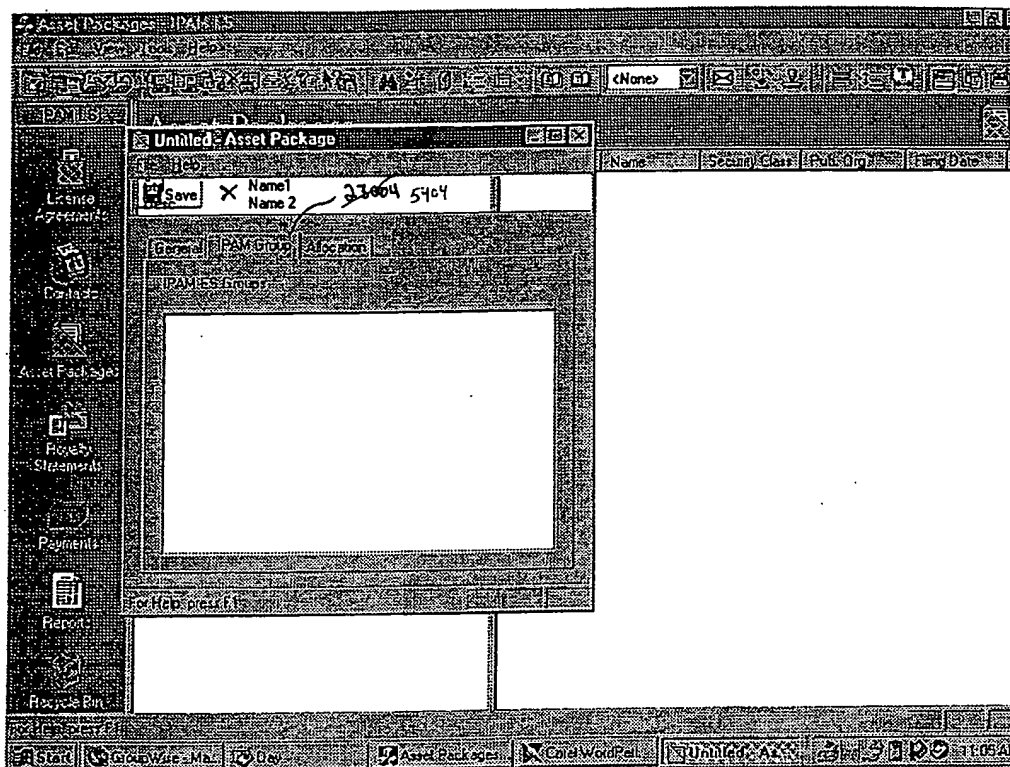
68/193

The screenshot shows a software window titled "Untitled - Asset Package". Inside, there's a "General" tab selected. The "Type" dropdown is set to "IPAM Group". The "Name" field is empty. The "Description" field is a large text area, also empty. Below it, the "Capture Period" is set to two empty date fields separated by "to". Above the window, there are handwritten notes: "X 5104 22704 23004 5404 22708 5108". To the right of the window, there's a handwritten note: "Group display 5402 23002".

5402  
~ Group display 23002

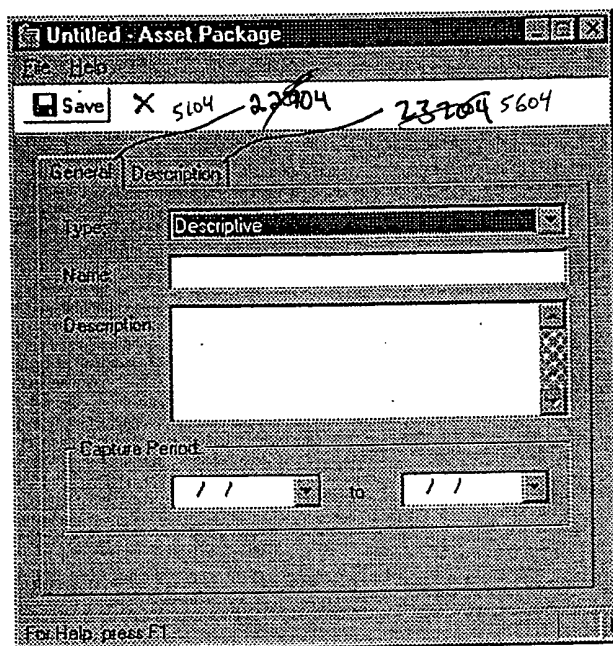
54  
FIG. 280

69/193



55  
FIL. 234

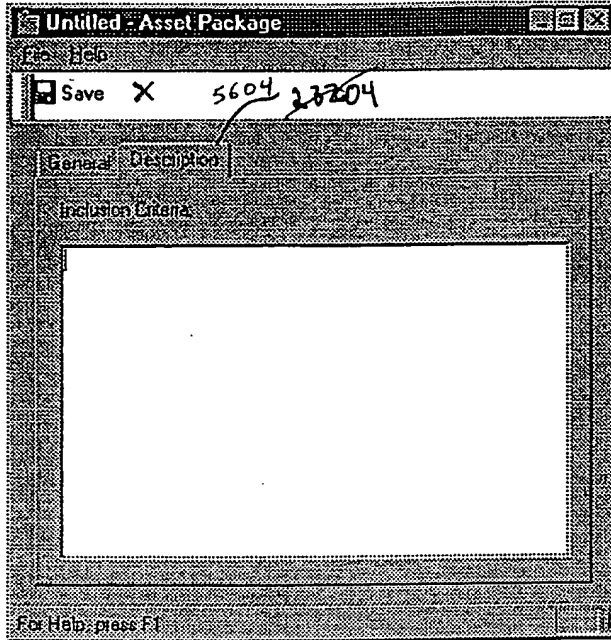
70/193



Descriptive Dialog, 5602, 23202

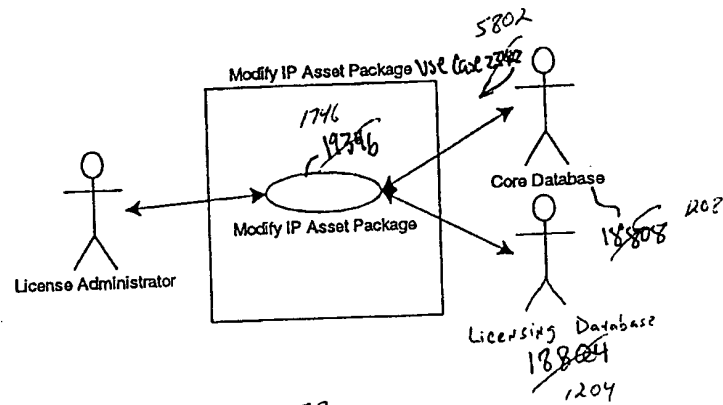
56  
FIG. 23/2

71/193



57  
FIG. 235

72/193



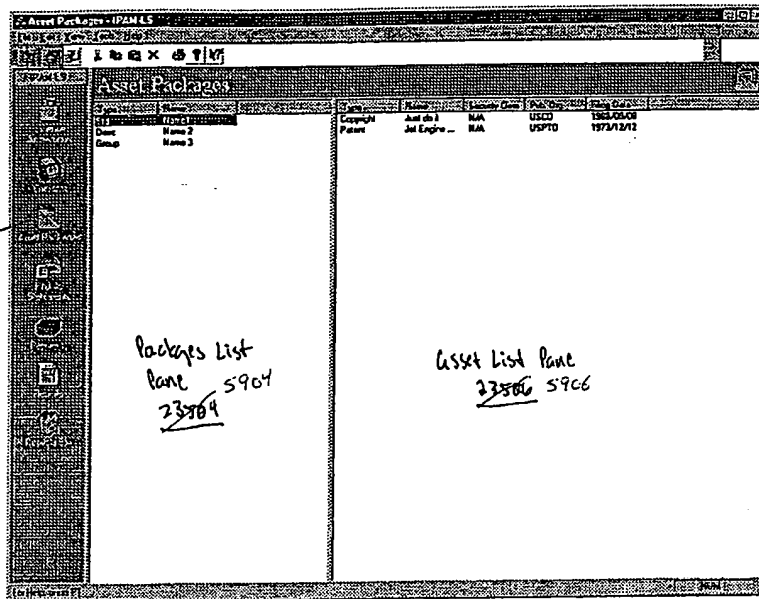
58  
FI 6.234



73/193

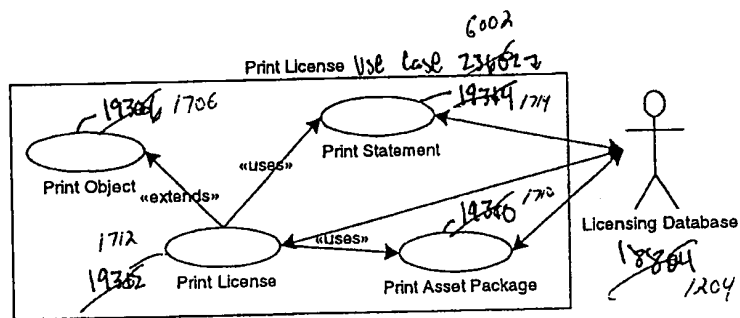
Asset Package View <sup>5902</sup>  
~~238027~~

Asset  
Package  
Icon  
~~19066~~  
1416



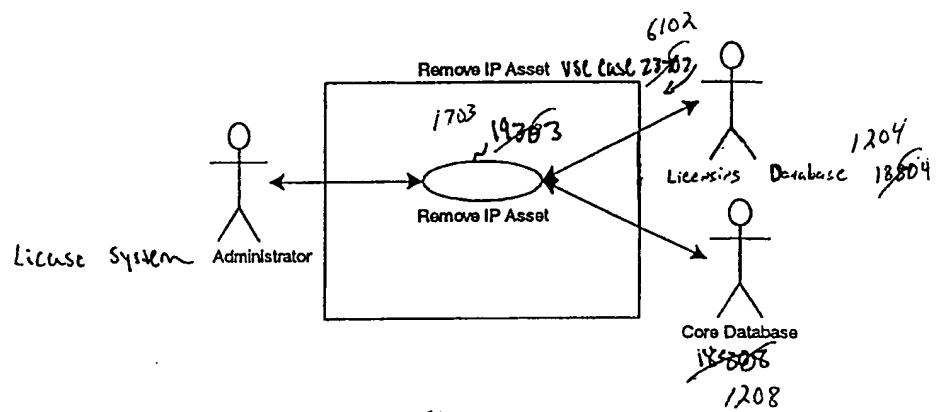
<sup>59</sup>  
FIG. 238

74/193



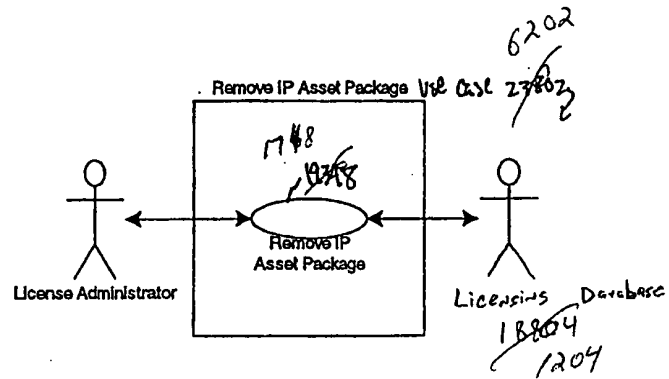
60  
Fig. 2386

75/193



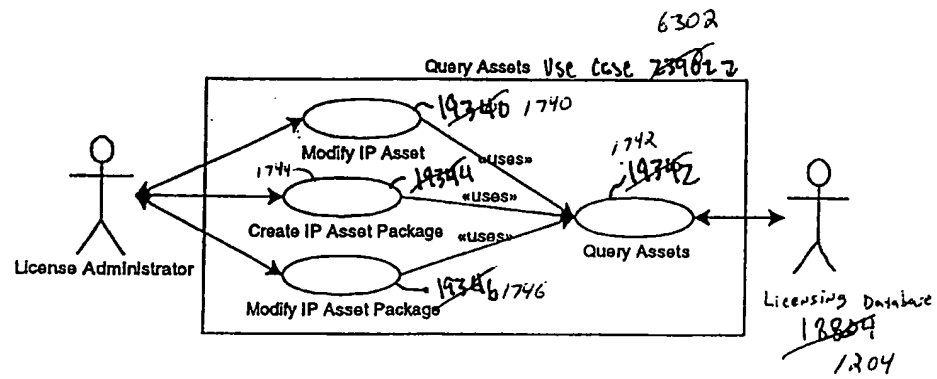
61  
FIG. 236

76/193



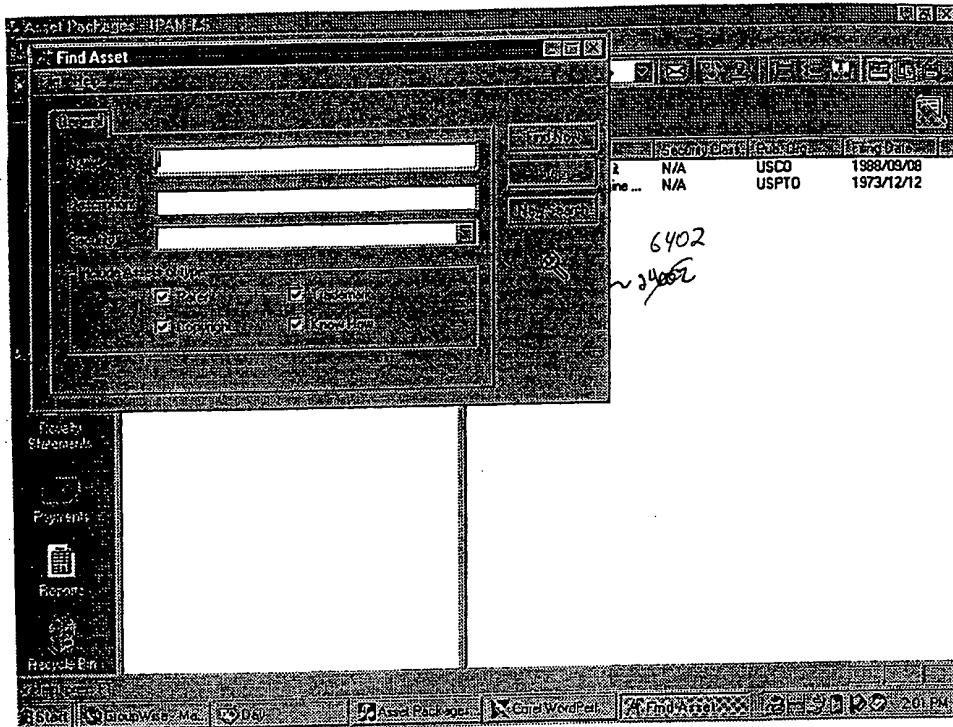
62  
FIG. 238

77/193



63  
FIG. 239

78/193



64A  
FIG. 24A

**Find Asset**

File Edit

Name:

Description:

Symbol:

Include Asset type:

☒ Patent ☒ Trademark

☒ Copyright ☒ Know How

Find Asset Stop New Search

Type	Asset
Patent	Asset 1
Trademark	Asset 2

6402

~24062

64B  
FIG. 2406

80/193

6504  
24104

6506  
24106

Find Asset

General Patent

Doc Number:

Title:

Filing Date:

Issue Date:

Pub. Org.:

Description:

Find Now

Stop

New Search

6502  
24102

65  
FIG. 2A



81/193

**Find Asset**

General Patent

Doc Number:

Title:


Filing Date:

Issue Date:

Pub. Off:

Description:

Find Next  
Stop  
New Search

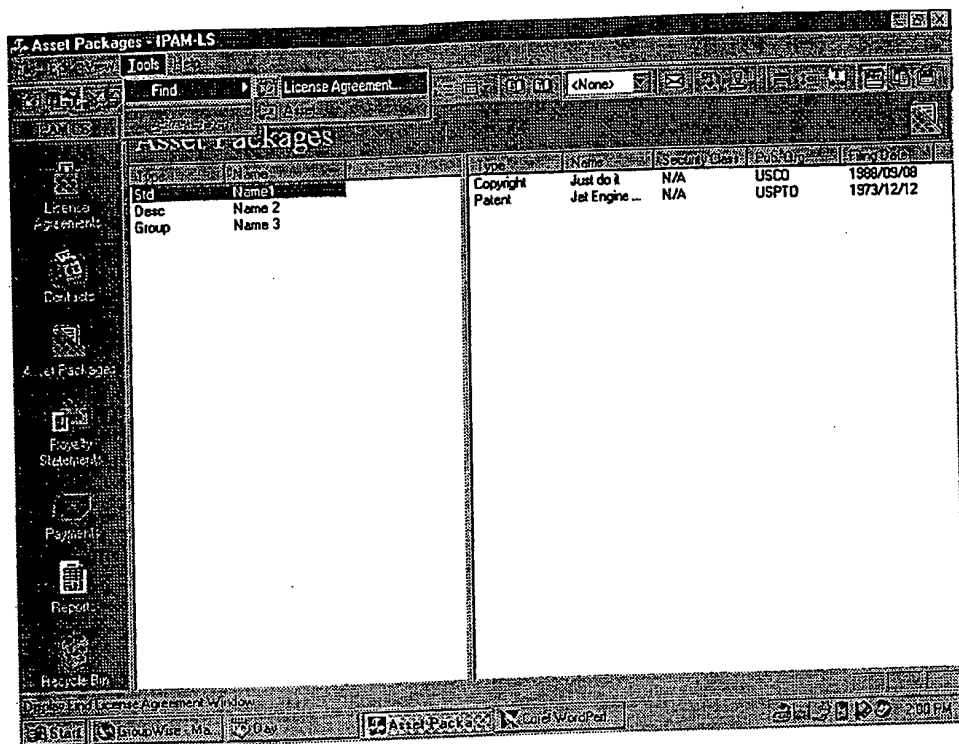


Type	Doc Number	Title	Filing Date	Issue Date
Patent	1234567	1234567	02/05/73	02/05/75
Copyright	453-A	Spam	11/12/87	12/12/88

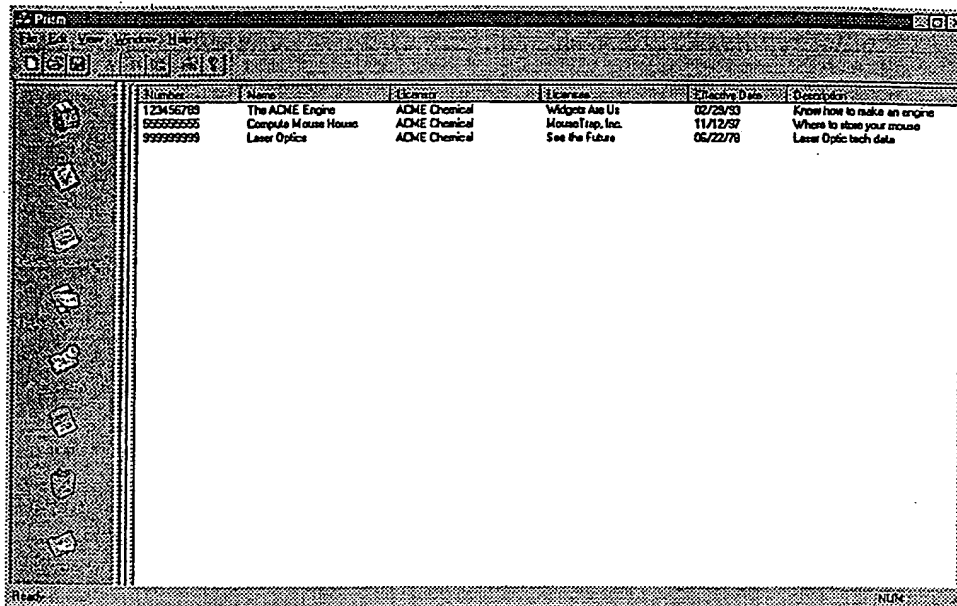
6602  
24282

66  
FIG. 249

82/193

67  
FIG. 2/3

83/193

6802  
License Agreements View 24482j

Number	Name	Company	License	Expiry Date	Description
123456789	The ADME Engine	ADME Chemical	Widgets Are Us	02/28/93	Know how to make an engine
999999999	Compute Mouse House	ADME Chemical	MouseTrap, Inc.	11/12/97	Where to store your mouse
999999999	Laser Optics	ADME Chemical	See the Future	05/22/78	Laser Optic tech data

68  
FIG. 244

84/193

Agreement Modification  
Dialog 24502 → 6902

The image shows a screenshot of a software dialog box titled "Agreement". The dialog box has a menu bar with "File", "View", and "Help". Below the menu bar is a toolbar with several icons. The main area of the dialog box is divided into several sections:

- General**: Contains fields for "Agreement ID" and "Title".
- Effective**: Contains two date fields separated by "to".
- Audit Window**: Contains two date fields separated by "to".
- Licensor**: A large text area for the licensor's information.
- Licensee**: A large text area for the licensee's information.
- Licensor Contacts**: A large text area for the licensor's contact information.
- Licensee Contacts**: A large text area for the licensee's contact information.
- Options**: A group of checkboxes including "Infringement", "Favored Nation", "Exclusive", "Sublicense", and "Assignable".
- Annual Guaranteed Minimum**: A text field for the annual guaranteed minimum.
- Description**: A large text area for the agreement's description.

Handwritten annotations are present throughout the image:

- 6904** and **24504** are written near the "General" section.
- 6906** and **24506** are written near the "Effective" section.
- 6908** and **24508** are written near the "Audit Window" section.
- 6910** and **24510** are written near the "Licensee" section.
- 6912** and **24512** are written near the "Licensee Contacts" section.
- 69** and **FIG. 245** are written at the bottom of the page.

7002  
24/02

Find License Agreement

General

ID:

Name:


Licensor:

Licensee:

Find Now

Stop

New Search



70  
F-IL-346

86/193

7102  
/ 24702

**Find License Agreement**

General

ID:

Name:


Licensor:

Licensee:

Find Name

Stop

New Search



ID	Name	Licensor	Lic
123456789	The ACME Engine	ACME Chemical	W
999999999	Laser Optics	ACME Chemical	Se

71  
FIL. 246

87/193

License Creation 7202  
Dialog 24802

6904  
24504

24506 6906  
24508 6908

**Agreement**

File View Help

General Asset Packages Terms

Agreement ID: [ ] Title: [ ]

Effective: [ ] to [ ] Audit Window: [ ] to [ ]

Licensor: [ ] Licensee: [ ]

Licensor Contacts: [ ] Licensee Contacts: [ ]

☐ Infringement ☐ Favored Nation  
☐ Exclusive ☐ Sublicense  
☐ Assignable

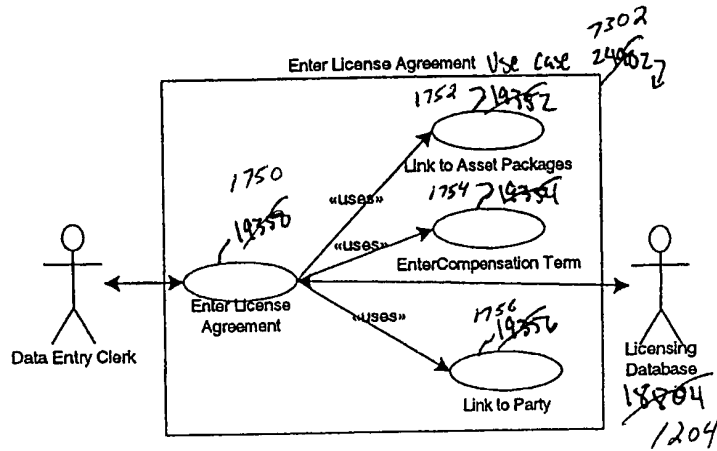
Annual Guaranteed Minimum: [ ]

Description: [ ]

Ready NUM

72  
FIG. 248

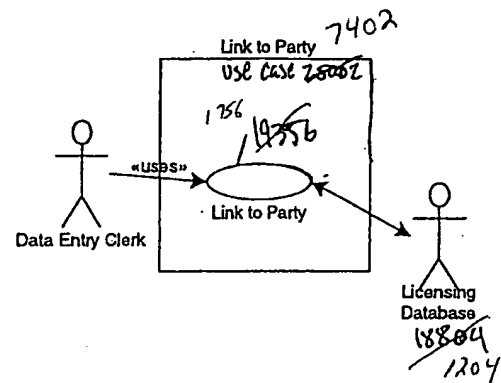
88/193



73  
FIG. 2A

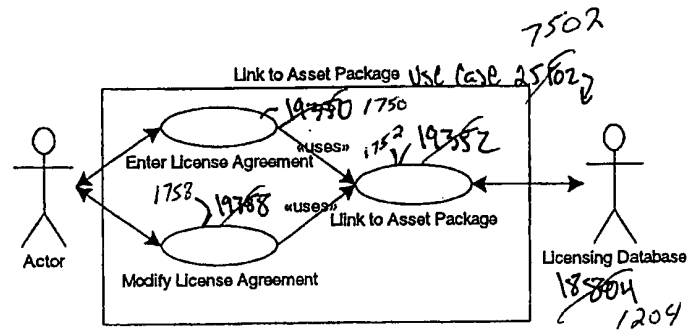


89/193



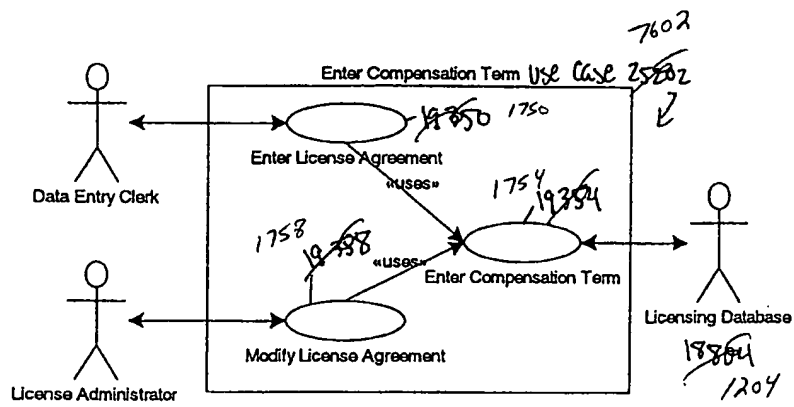
74  
FIG. 280

90/193



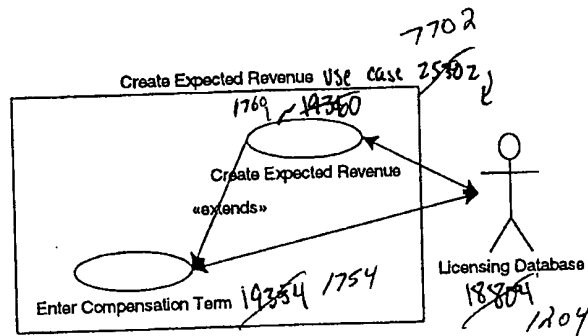
75  
FIG. 28

91/193



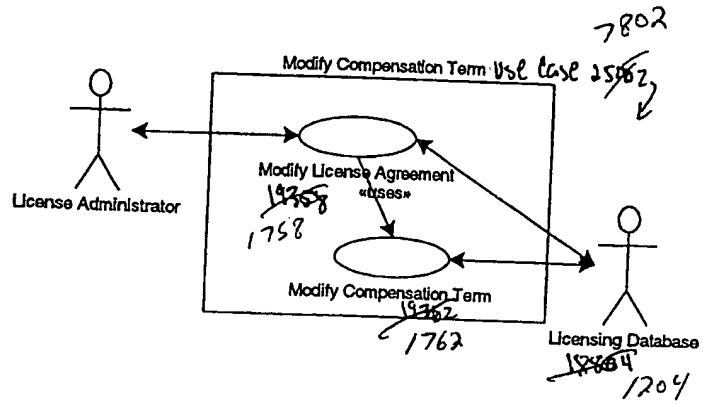
76  
FIG. 2582

92/193



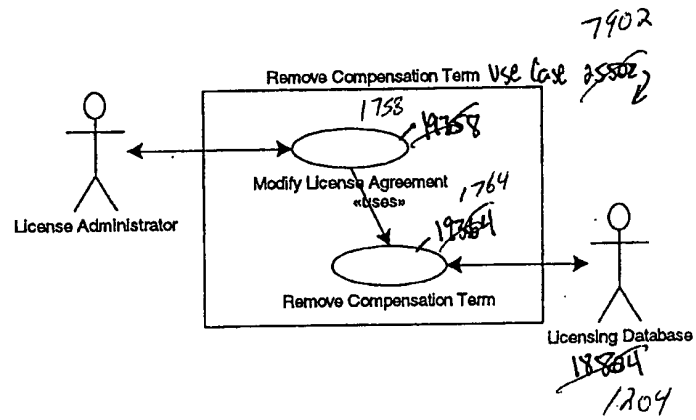
77  
FIG. 253

93/193



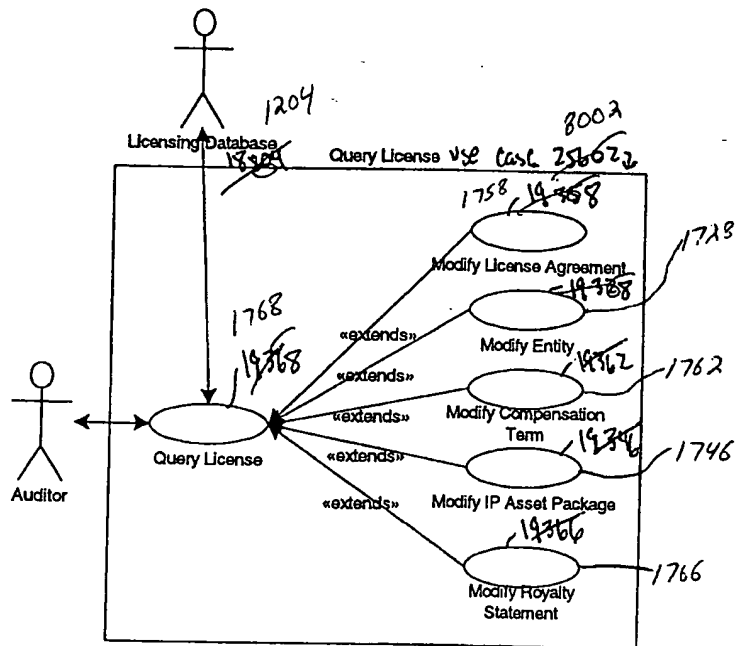
78  
FIG. 284

94/193



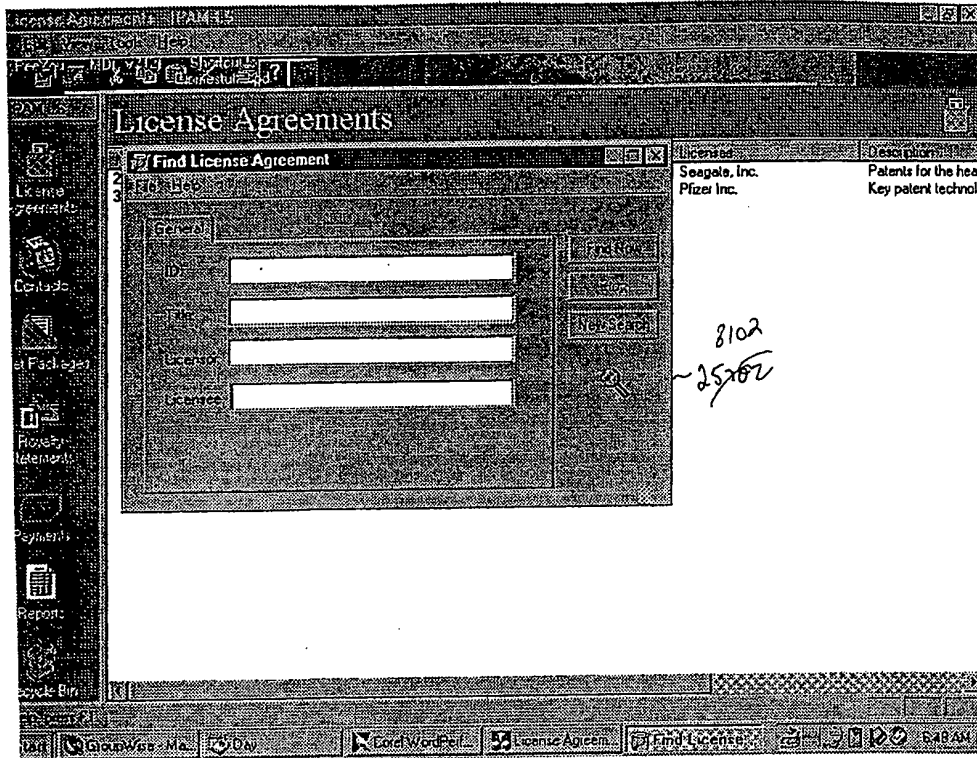
79  
FIG. 258

95/193



80  
FIG. 280

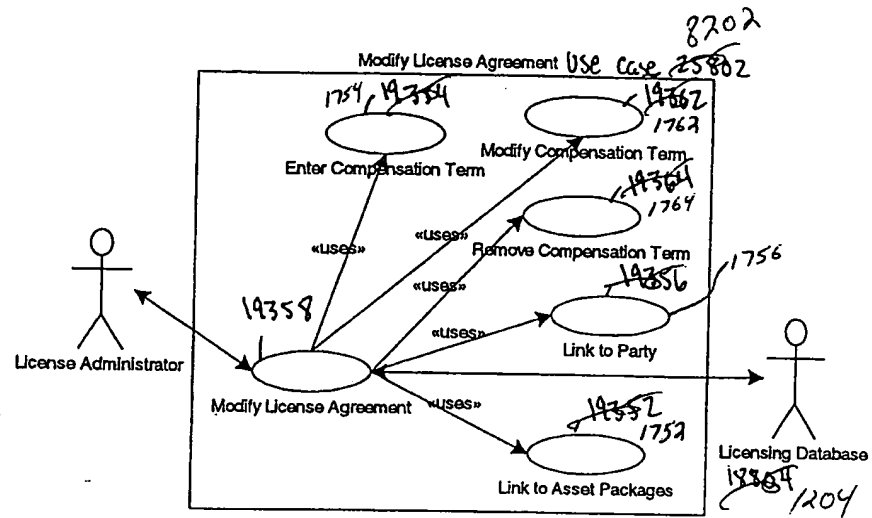
96/193



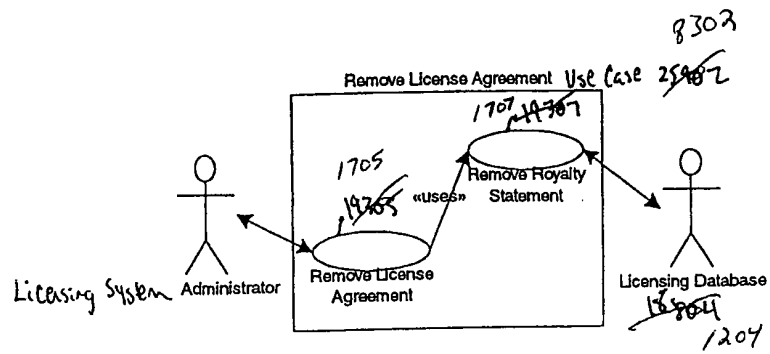
81  
FIG. 1



97/193

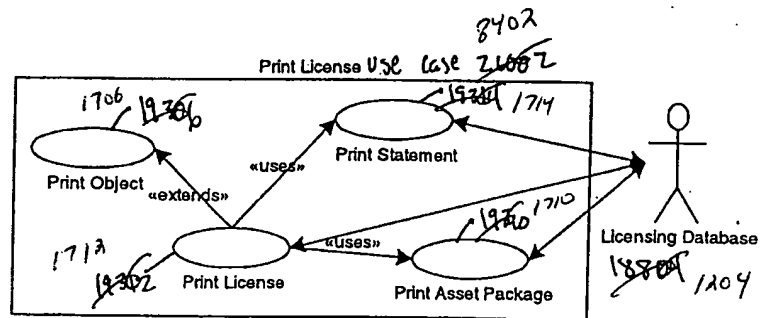
82  
FIG. 288

98/193



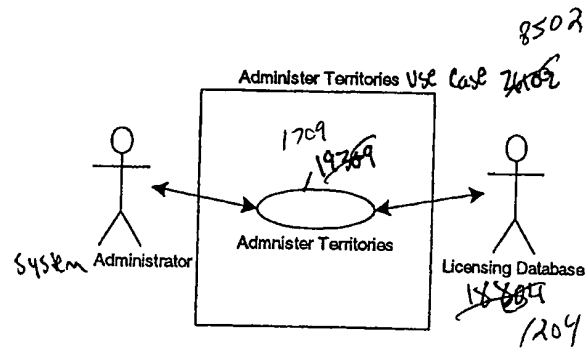
83  
FIG. 289

99 / 193



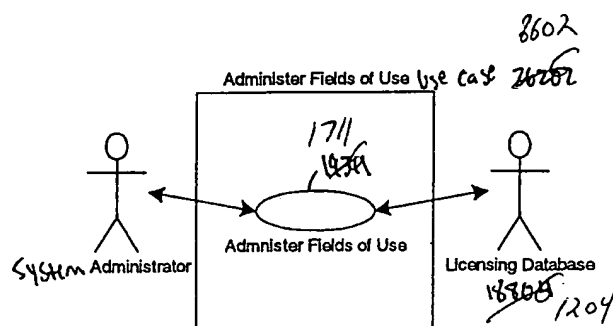
84  
FIG. 266

100/193



85  
FIG. 26A

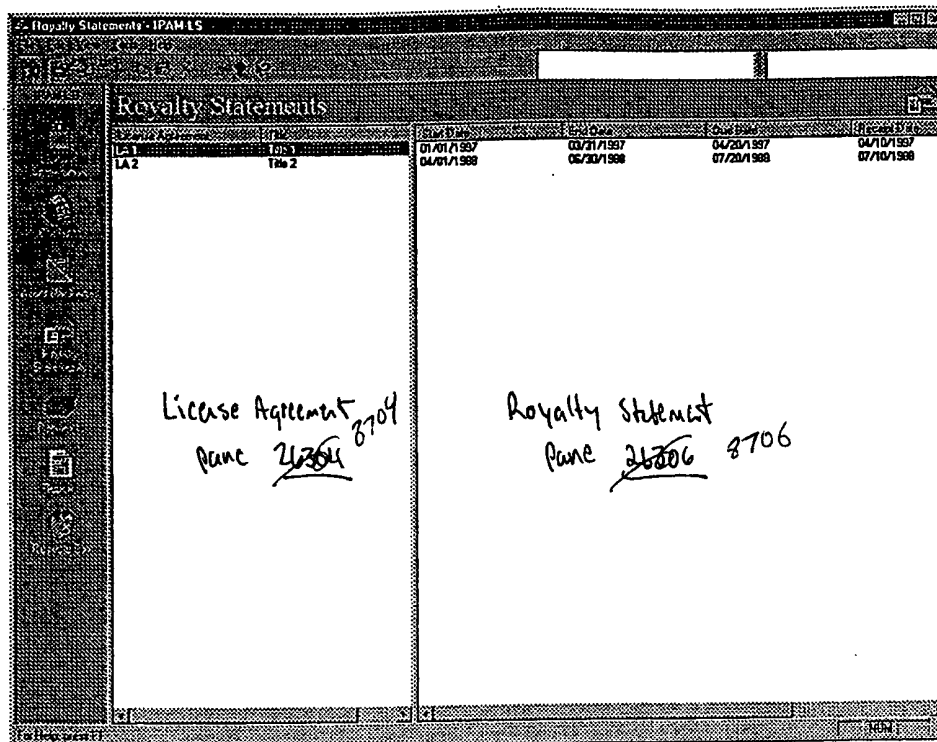
101/193



86  
FIG. 262

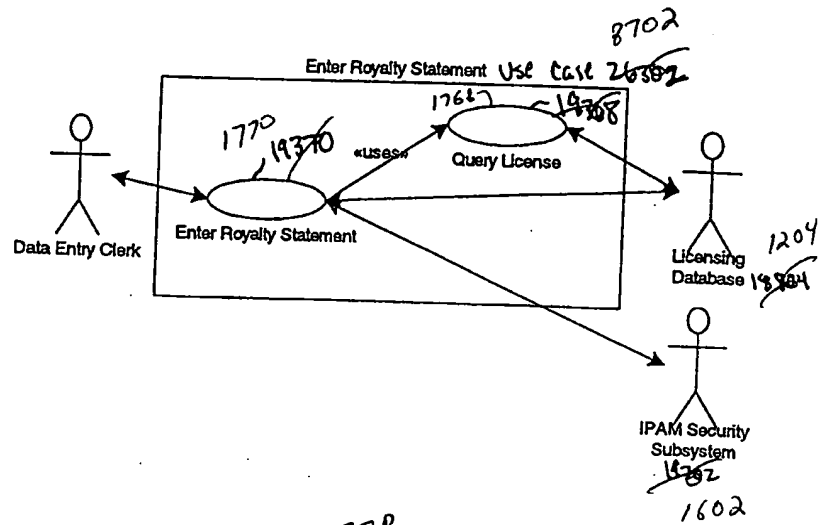
102/193

Royalty statement view ~~26302~~ 879

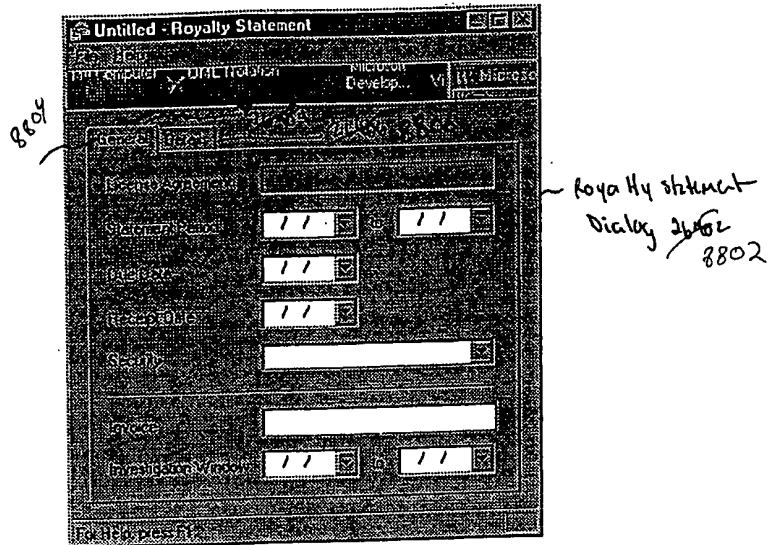


87A  
FIG. 263A

103/193



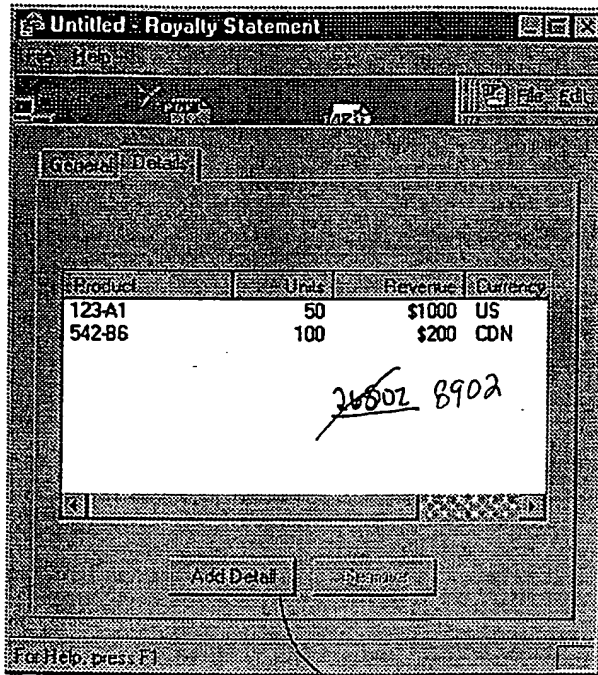
104/193



88  
FIG. 2602



105/193



89  
FIG. 265

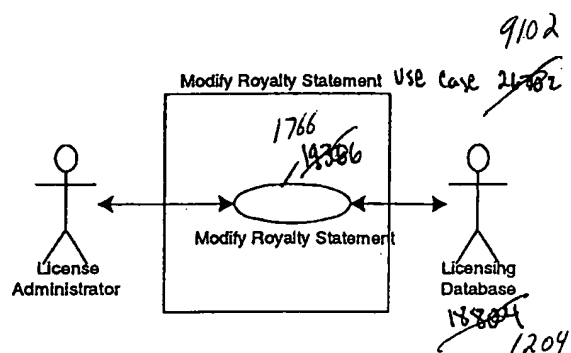
2604 8904

106/193

9002  
Detail Dialog 2602

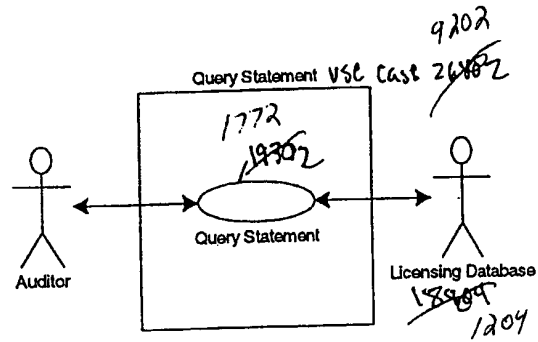
90  
FIG. 216

107/193



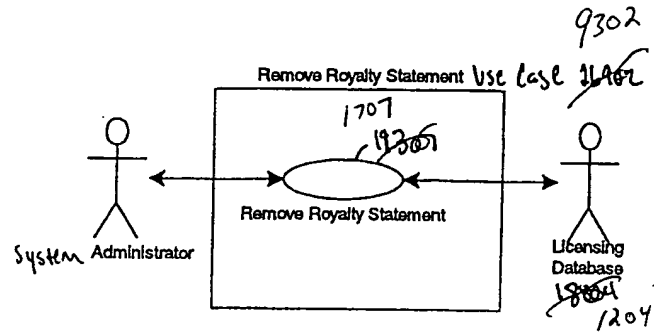
91  
FIG. 26

108/193



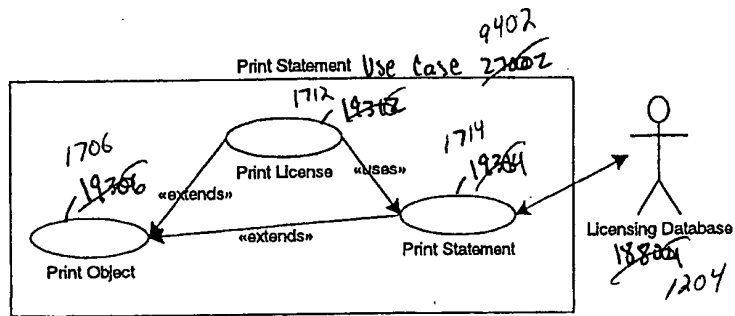
92  
FIG. 268

109/193



93  
FIG. 26A

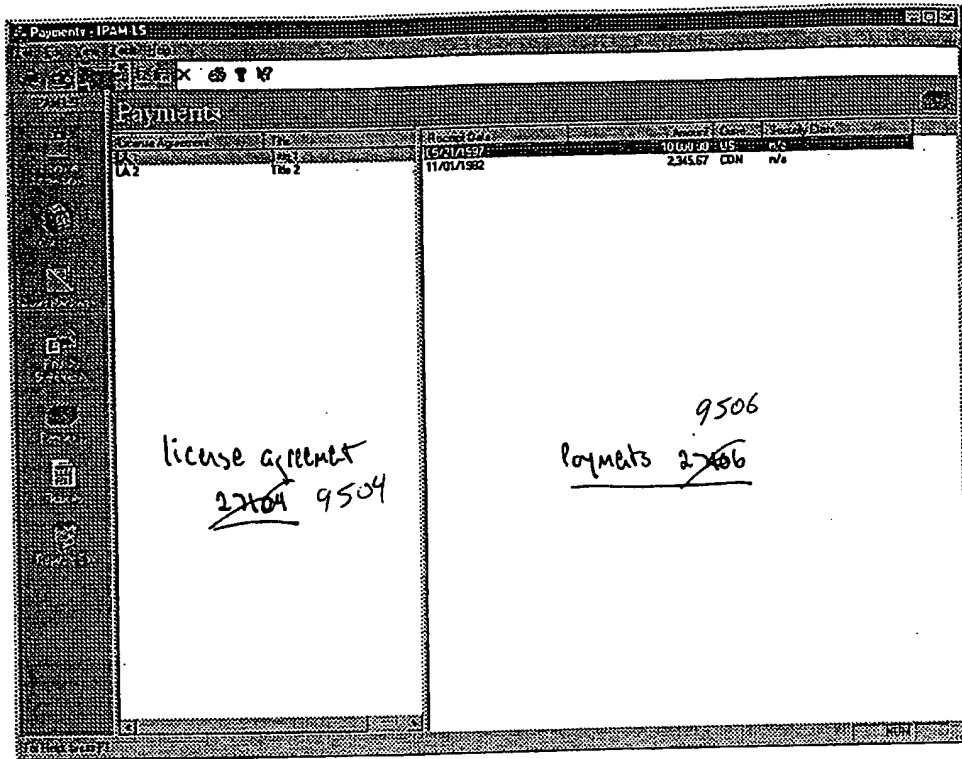
110/193



94  
FIG. 2/90

111 / 193

9502  
payments View ~~2762~~



95  
FIG. 276

112/193

9604  
~2284

06/10/1998 - Payment

Requester's Name: [Redacted]

Request Date: 6/10/1998

Amount: [Redacted]

Specify Date: [Redacted]

Options:

Loan Payment: [Redacted]

Amount: [Redacted]

INUM1

9602  
~27802

96  
FIG. 272



113/193

06/10/1998 - Payment

Save X 9606

General Royalty Statement Detail Expected Revenue

Description	Period	Amount	Outstanding	Amount to Pay
12321	01/01/1998 - 03/32/1998	\$100.00	\$0.00	\$0.00
45A1		\$60.00	\$0.00	\$0.00
6286		\$40.00	\$0.00	\$0.00
12321	04/01/1998 - 06/32/1998	\$80.00	\$0.00	\$0.00
45A1		\$50.00	\$0.00	\$0.00
S4-C0		\$30.00	\$0.00	\$0.00

Amount Allocated: 0.00

Payment Total: 180.00

Percent Allocated: 0%

For Help, press F1

NUM

2780 9702

2780 9704

2780 9706

2780 9708

2780 9702

2780 9704

2780 9706

2780 9708

97  
FIG. 278

114/193

06/10/1998 - Payment

Save X

9608

General Policy Statement Detail Expected Revenue

Term	Period	Expected	Actual
10% of Revenue	04/01/199...	\$80.00	\$0.00
Annual Fee	01/01/199...	\$100.00	\$0.00

Term pane 9810 27460

Estimate pane 9808 27468

Amount Allocated: 0.00

Payment Total: 0.00

Percent Allocated: 0%

9802 27462 9806 27486 9804 27404

For Help, press F1

NUM

98  
FIG. 289

115/193

06/10/1998 - Payment

Save X

General Royalty Statement Details Expected Revenue

9608

10% of Revenue

Annual Fee

Period	Expected	Actual
1998	\$2000.00	\$0.00

Amount Allocated: 0.00

Payment Total: 0.00

Percent Allocated: 0%

For Help, press F1

NUM

99

Fig. 205

116/193

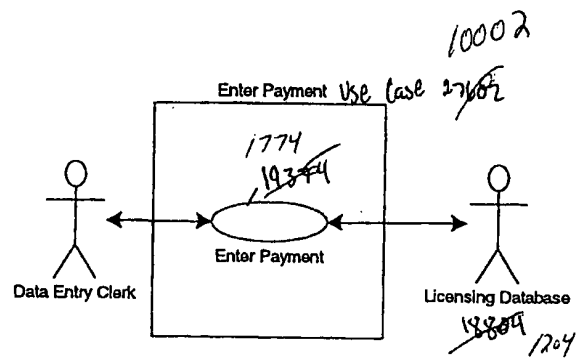
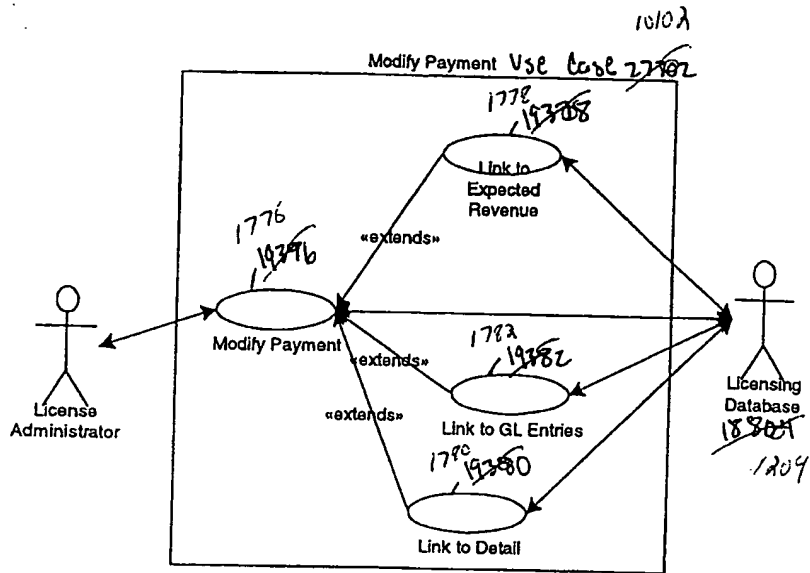


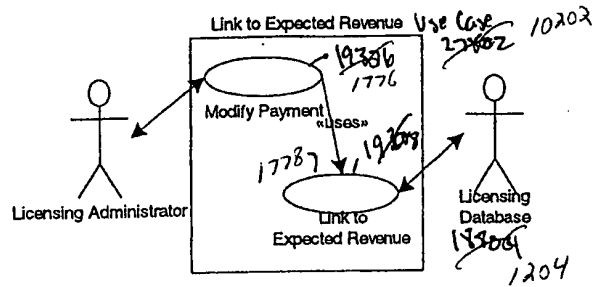
FIG. 2 <sup>100</sup> ~~276~~

117/193



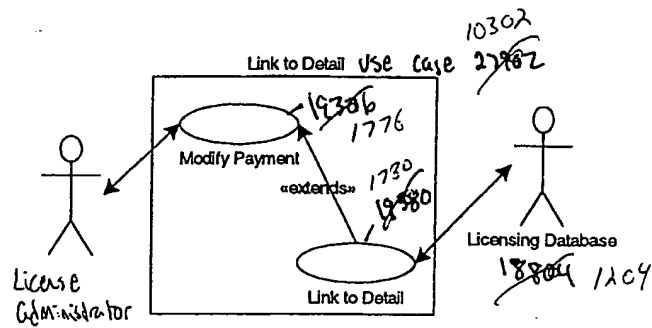
101  
FIG. 271

118/193



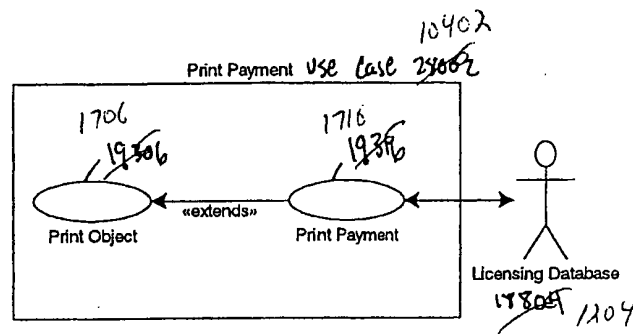
102  
FIG. 278

119/193



103  
FIG. 279

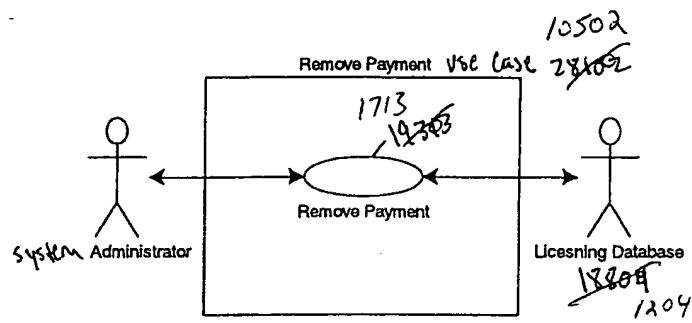
120/193



104  
FIG. 280



121/193



105  
FIG. 289

122/193

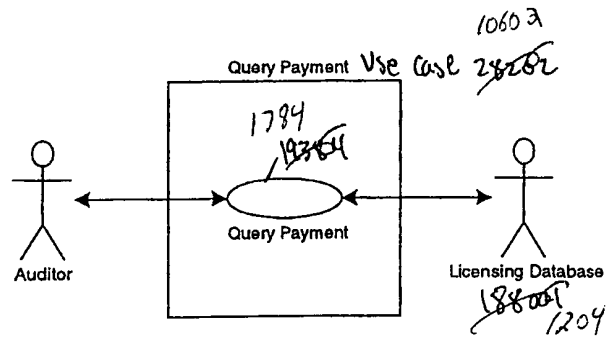
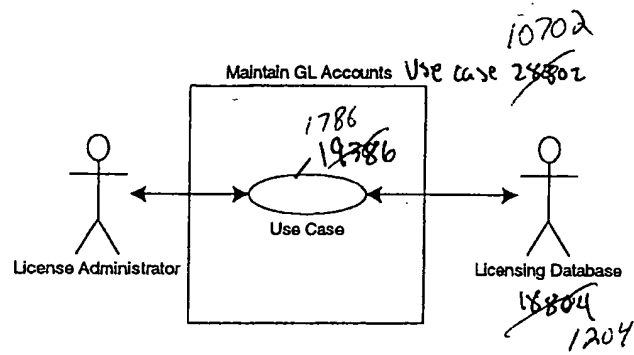


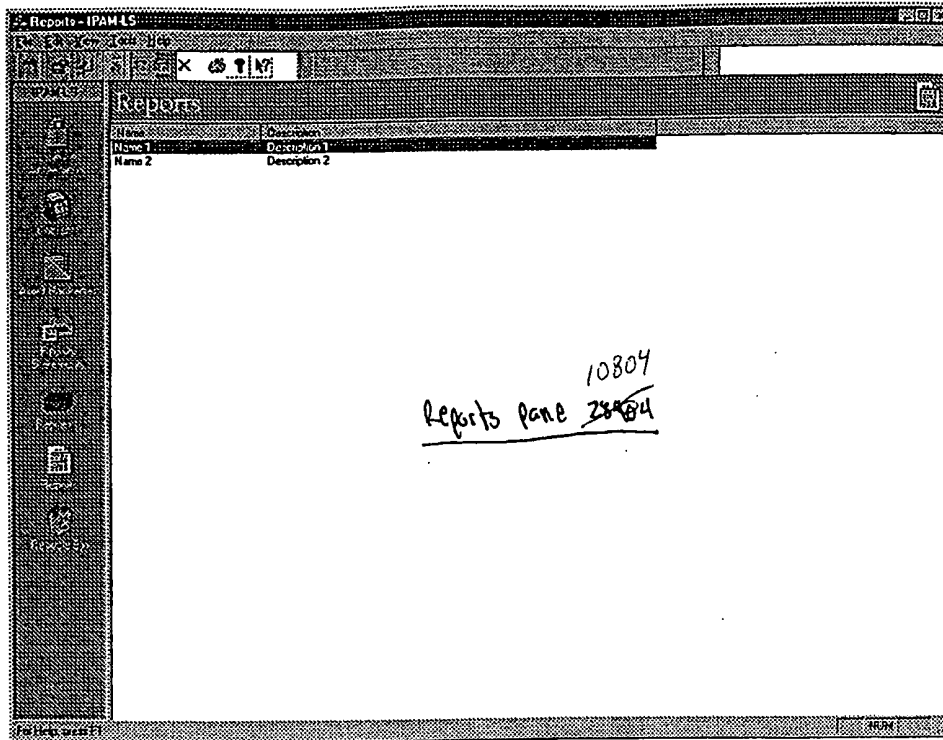
FIG. 282



107  
FIL. 283

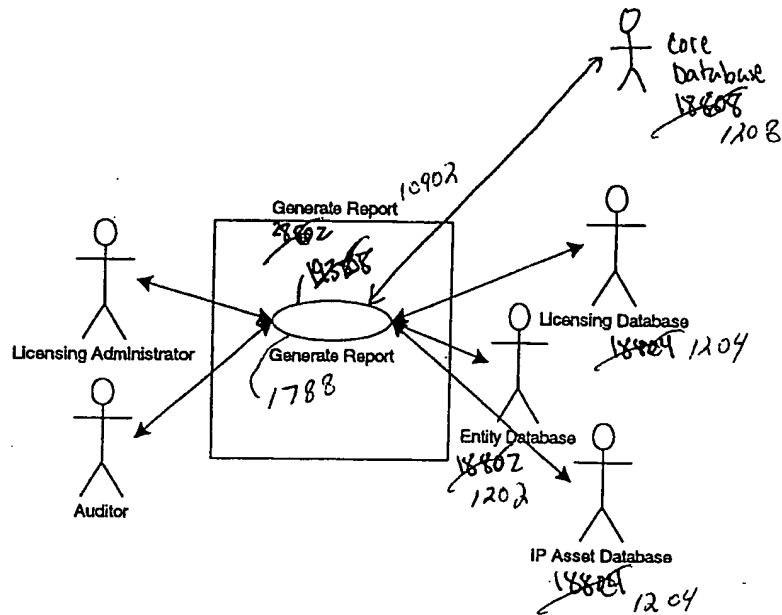
124/193

Reports View ~~28402~~ 10802



108  
FIG. 284

125/193



109  
FIG. 286

126/193

## Payment Exception Report

08/10/1998	Effective Date	Expiration Date	Agreement Type
LA145433556	04/15/1996	04/15/2010	License
Payment Information			
LA145433556 Molecular-Bio USPTO 5597619	Expected License Revenue		
Receipt Date	Due Date	Expected Amt	Late Fee
01/01/1998	01/01/1998	\$20,000.00	\$0.00
02/01/1998	01/01/1998	\$1,000.00	\$0.00
05/01/1998	04/01/1998	\$2,000.00	\$0.00
		\$23,000.00	\$0.00
Payment Information			
LA145433557 Molecular-Bio EP B 560556	Expected License Revenue		
Receipt Date	Due Date	Expected Amt	Late Fee
08/10/1998	08/05/1998	\$15,000.00	\$0.00
02/01/1998	01/01/1998	\$1,000.00	\$0.00
05/01/1998	04/01/1998	\$2,000.00	\$0.00
		\$18,000.00	\$0.00
Payment Information			
LA145433558 Molecular-Bio Flu Mask Know How	Expected License Revenue		
Receipt Date	Due Date	Expected Amt	Late Fee
03/20/1998	03/15/1998	\$30,000.00	(\$2,000.00)
02/01/1998	01/01/1998	\$1,000.00	\$0.00
05/01/1998	04/01/1998	\$2,000.00	\$0.00
		\$33,000.00	(\$2,000.00)
Payment Information			
LA145433559 Molecular-Bio Ebolus Program	Expected License Revenue		
Receipt Date	Due Date	Expected Amt	Late Fee
06/15/1998	06/15/1998	\$20,000.00	\$0.00
02/01/1998	01/01/1998	\$1,000.00	\$0.00
05/01/1998	04/01/1998	\$2,000.00	\$0.00
		\$23,000.00	\$0.00

110A

FLB-786A

5

Copyright © 1998 by SmartPatents, Inc. SmartPatents, Inc. Confidential. Do not duplicate.

128/193

# Payment Exception Report

08/10/1998

Payment Exception Report

Agreement ID Name LA145433564 E

Effective Date Expiration Date Agreement Type

LA145433564 BioSpecifics USPTO 5597619

04/15/1996 04/15/2010 License

Payment Information

Expected License Revenue

Receipt Date Payment Amt Withheld Amt Allocation Amt Due Date Expected Amt Late Fee Balance Amt

05/15/1998 \$5,000.00 \$5,000.00 05/13/1998 \$5,000.00 \$0.00

02/01/1998 \$500.00 \$500.00 01/01/1998 \$1,000.00 (\$500.00)

05/01/1998 \$2,000.00 \$2,000.00 04/01/1998 \$2,000.00 \$0.00

\$7,500.00 \$8,000.00 (\$500.00)

Grand Total: \$12,500.00

LA145433565 Argen USPTO 5597619

04/15/1996 04/15/2010 License

Payment Information

Expected License Revenue

Receipt Date Payment Amt Withheld Amt Allocation Amt Due Date Expected Amt Late Fee Balance Amt

02/01/1998 \$1,000.00 \$1,000.00 01/01/1998 \$1,000.00 \$0.00

05/01/1998 \$2,000.00 \$2,000.00 04/01/1998 \$2,000.00 \$0.00

\$3,000.00 \$3,000.00 \$0.00

Grand Total: \$12,500.00

LA145433565 Amgen USPTO 5597619

08/10/1998										Agreement ID										Name										Effective Date										Expiration Date										Agreement Type																													
										LA145433565										Amgen USPTO 5597619										04/15/1996										04/15/2010										License																													
Payment Information																																																																															
Receipt Date										Payment Amt										Withheld Amt										Allocation Amt										Due Date										Expected Amt										Late Fee										Balance Amt									
02/01/1998										\$1,000.00																				\$1,000.00										01/01/1998										\$1,000.00																				\$0.00									
05/01/1998										\$2,000.00																				\$2,000.00										04/01/1998										\$2,000.00																				\$0.00									
																														\$3,000.00																				\$3,000.00																				\$0.00									
Grand Total:																														\$221,500.00																				\$234,000.00																				(\$12,500.00)									

110C  
FIG. 286C



129/193

## IP Asset - Agreement Financial Detail

06/10/1998

Package ID Package Name

Description

2

US 5597619

Effective Date Expiration Date Agreement Type

LA145433556

Molecular-Bio USPTO 5597619

Company A

Receipt Date  
01/01/1998  
02/01/1998  
05/01/1998

Amount Allocated  
\$20,000.00  
\$1,000.00  
\$2,000.00  
\$23,000.00

Due Date  
01/01/1998  
01/01/1998  
04/01/1998

Expected Revenue  
\$20,000.00  
\$1,000.00  
\$2,000.00  
\$23,000.00

LA145433564

BioSpecifics USPTO 5597619

Company B

Receipt Date  
05/15/1998  
02/01/1998  
05/01/1998

Amount Allocated  
\$5,000.00  
\$500.00  
\$2,000.00  
\$7,500.00

Due Date  
05/13/1998  
01/01/1998  
04/01/1998

Expected Revenue  
\$5,000.00  
\$1,000.00  
\$2,000.00  
\$8,000.00

LA145433565

Amgen USPTO 5597619

Company C

Receipt Date  
02/01/1998  
05/01/1998

Amount Allocated  
\$1,000.00  
\$2,000.00  
\$3,000.00  
\$34,000.00

Due Date  
01/01/1998  
04/01/1998

Expected Revenue  
\$1,000.00  
\$2,000.00  
\$3,000.00  
\$34,000.00

3

EP B 560556

Effective Date Expiration Date Agreement Type

LA145433557

Molecular-Bio EP B 560556

Company A

Receipt Date  
06/10/1998  
02/01/1998  
05/01/1998

Amount Allocated  
\$15,000.00  
\$1,000.00  
\$2,000.00  
\$18,000.00

Due Date  
06/05/1998  
01/01/1998  
04/01/1998

Expected Revenue  
\$15,000.00  
\$1,000.00  
\$2,000.00  
\$18,000.00

Copyright © 1998 by SmartPatents, Inc. SmartPatents, Inc. Confidential. Do not duplicate.

130/193

9562-911  
1118

Copyright © 1998 by SmartPatents, Inc. SmartPatents, Inc. Confidential. Do not duplicate.

131/193

111C

FIG. 291C

## IP Asset - Agreement Financial Detail

08/10/1998

Package ID      Package Name

Description

6      Roche Molecular Logo

Agreement ID	Name	Effective Date	Expiration Date	License Due Date	Expected Revenue	Agreement Type
LA145433560	Molecular Bio Roche Molecular Logo <i>Company A</i>	04/15/1996	04/15/2010	01/20/1998	\$10,000.00	
		Amount Allocated			\$1,000.00	
				01/01/1998	\$1,000.00	
				04/01/1998	\$2,000.00	
					\$13,000.00	
					\$13,000.00	

7      All Patents

Agreement ID	Name	Effective Date	Expiration Date	License Due Date	Expected Revenue	Agreement Type
LA145433560	Molecular Bio Roche Molecular Logo <i>Company A</i>	04/15/1996	04/15/2010	01/20/1998	\$10,000.00	
		Amount Allocated			\$1,000.00	
				01/01/1998	\$1,000.00	
				04/01/1998	\$2,000.00	
					\$13,000.00	
LA145433561	Pfizer All Patents <i>Company D</i>	04/15/1996	04/15/2010	01/20/1998	\$1,000.00	
		Amount Allocated			\$2,000.00	
				04/01/1998	\$3,000.00	
					\$15,000.00	

Copyright © 1998 by SmartPatents, Inc. SmartPatents, Inc. Confidential. Do not duplicate.

3

132/193

111D  
F16.2875

## IP Asset - Agreement Financial Detail

Package ID	Package Name	Description		
06/10/1998	Chemotherapy Application Know How			
8				
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type
LA145433563	Return-Med Chemotherapy Tech Data <i>Confidential</i>	04/15/1998	04/15/2010	License
		Amount Allocated	Due Date	Expected Revenue
		\$50,000.00	06/06/1998	\$50,000.00
		\$1,000.00	01/01/1998	\$1,000.00
		\$2,000.00	04/01/1998	\$2,000.00
		\$53,000.00		\$53,000.00
		\$53,000.00		\$53,000.00
		\$233,500.00		\$247,000.00
Grand Total:				

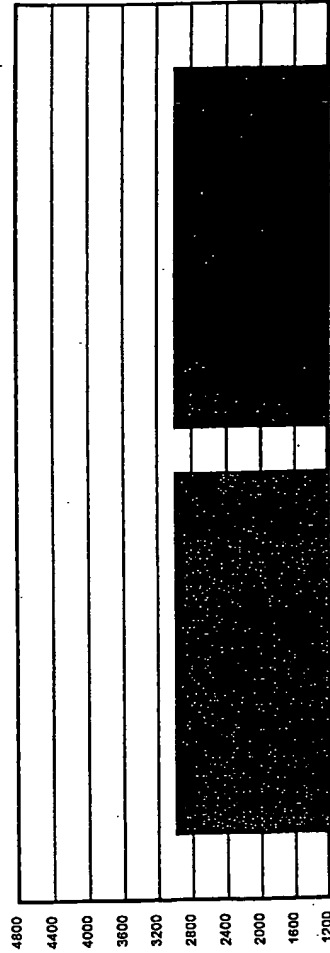
112A  
FIG. 286A

# Licensee Profile

06/10/1998

Amgen, Inc.  
Company A

■ Allocated Payments (USD)  
■ Expected Revenue (USD)



LA145433565

Agreement ID Name  
LA145433565 Amgen USPTO 5597619  
Company A

Effective Date Expiration Date Agreement Type  
04/15/1996 04/15/2010 License

134/193

112B  
FIG. 2A

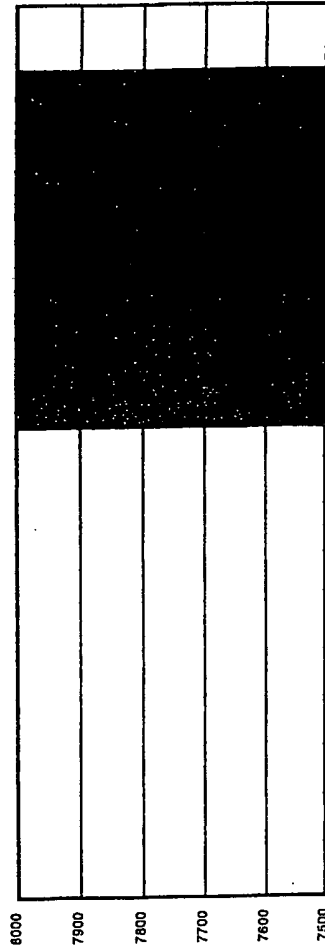
# Licensee Profile

06/10/1998

BioSpecifics Technologies Corp.

Confidential

■ Allocated Payments  
■ Expected Revenue



LA145433564

Agreement ID Name  
LA145433564 BioSpecifics USPTO 5597619

Confidential

Effective Date Expiration Date Agreement Type  
04/15/1996 04/15/2010 License

135/193

113C  
FI 288

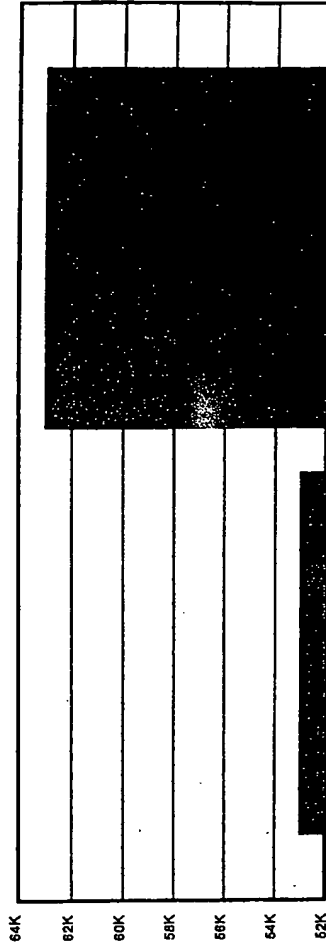
# Licensee Profile

08/10/1998

Future Medical Technologies, Inc.

Company C

■ Allocated Payments  
■ Expected Revenue



LA145433563

Effective Date 04/15/1996  
Expiration Date 04/15/2010  
Agreement Type License

Agreement ID LA145433563  
Name Future Med Chemotherapy Tech Data  
Company C

136/193

11AD  
Fil. 285

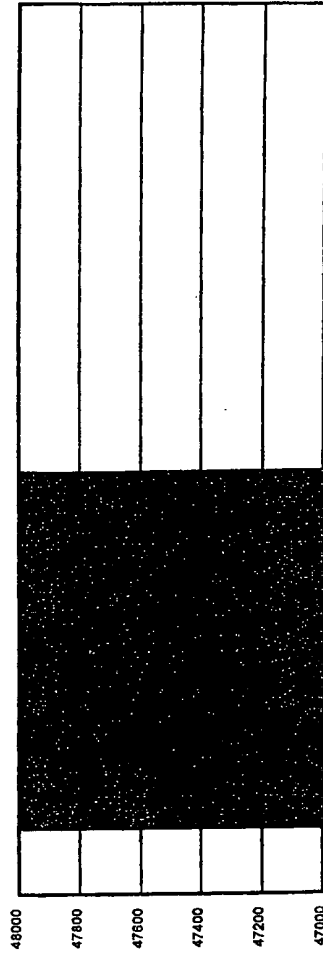
# Licensee Profile

08/10/1998

Life Medical Sciences Corp.

Company D

■ Allocated Payments  
■ Expected Revenue



LA145433562

Agreement ID LA145433562  
Name EferMed Thru Mask Know How  
Company D

Effective Date 04/15/1996  
Expiration Date 04/15/2010  
Agreement Type License



137/193

112E  
Fig. 38E

# Licensee Profile

08/10/1998

Molecular Biosystems, Inc.

Company E

■ Allocated Payments  
■ Expected Revenue



Agreement ID	Name	Effective Date	Expiration Date	Agreement Type
LA145433556	Molecular Bio USPTO 6597619	04/15/1996	04/15/2010	License
LA145433557	Molecular Bio EP B 560556	04/15/1996	04/15/2010	License
LA145433558	Molecular Bio Flu Mask Know How	04/15/1996	04/15/2010	License
LA145433559	Molecular Bio Ebola Program	04/15/1996	04/15/2010	License
LA145433560	Molecular Bio Roche Molecular Logo	04/15/1996	04/15/2010	License

138/193

11X.F  
Fig. 238F

Licensee Profile

06/10/1998

Pfizer Inc: *Company F*

■ Allocated Payments  
■ Expected Revenue



Effective Date 04/15/1996  
Expiration Date 04/15/2010  
Agreement Type License

Agreement ID LA145433561  
Name Pfizer All Patents  
*Company F*

139/193

## Agreement Summary

Fig. 289

06/10/1998

Agreement ID	Name	Type
LA145433556	Molecular Bio USPTO 5597619	License

Effective Date	Expiration Date
04/15/1996	04/15/2010

Exclusivity	<input type="checkbox"/>
Assignabl	<input type="checkbox"/>
Transferabl	<input type="checkbox"/>
Revocabl	<input type="checkbox"/>
Confidenti	<input checked="" type="checkbox"/>

Description

## Agreement Parties

Name	Role Name	Description
Molecular Biosystems, Inc. @ Company A	Licensee	0-0-0 ext. 0 Fax
Roche Molecular Systems @ Company B	Licensor	0-0-0 ext. 0 Fax

## Compensation Terms

Description	Late Fee	Amount	Term Type
		\$20,000.00	F
		\$0.00	R

140/193

Licensee Asset Package Summary					
06/10/1998					
<b>Amgen, Inc. Company A</b>					
Package Name	Description			Package Type	
US 5597619				Standard	
Asset	Asset Type				
USPTO 5597619	USPTO Document				
License Agreements					
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type	
LA145433565	Amgen USPTO 5597619 Company A	04/15/1996	04/15/2010	License	
<b>BioSpecifics Technologies Corp. Company B</b>					
Package Name	Description			Package Type	
US 5597619				Standard	
Asset	Asset Type				
USPTO 5597619	USPTO Document				
License Agreements					
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type	
LA145433564	BioSpecifics USPTO 5597619 Company B	04/15/1996	04/15/2010	License	
<b>Eutune Medical Technologies Inc. Company C</b>					
Package Name	Description			Package Type	
Chemotherapy Application Know How				Standard	
Asset	Asset Type				
	Technical Data				
License Agreements					
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type	
LA145433563	Eutune-Med-Chemotherapy Tech Data Company C	04/15/1996	04/15/2010	License	
<b>Life-Medical Sciences Corp. Company D</b>					
Package Name	Description			Package Type	
Influenza Sanitary Mask Know How				Standard	
Asset	Asset Type				
USPTO 5597619	USPTO Document Technical Data				
License Agreements					
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type	
LA145433562	Life-Med-Flu-Mask Know How Company D	04/15/1996	04/15/2010	License	
Copyright © 1998 by SmartPatents, Inc. SmartPatents, Inc. Confidential. Do not duplicate.					

 File 296  
 114A

## Licensee Asset Package Summ:

06/10/1998

Molecular Biosystems, Inc. *Company E*

Package Name	Description	Package Type
All Patents		Standard
Asset	Asset Type	
USPTO 5597619	USPTO Document	
USPTO 32028	USPTO Document	
EPO B 560556	EPO Document	
License Agreements		
Agreement ID	Name	Effective Date Expiration Date Agreement Type
LA145433560	<i>Company E</i> Molecular Bio Roche Molecular Logo	04/15/1996 04/15/2010 License
Package Name	Description	Package Type
Anti-Ebolus Vaccine and Diagnost Test		Standard
Asset	Asset Type	
	Copyright	
License Agreements		
Agreement ID	Name	Effective Date Expiration Date Agreement Type
LA145433559	<i>Company E</i> Molecular Bio Ebolus Program	04/15/1996 04/15/2010 License
Package Name	Description	Package Type
EP B 560556		Standard
Asset	Asset Type	
EPO B 560556	EPO Document	
License Agreements		
Agreement ID	Name	Effective Date Expiration Date Agreement Type
LA145433557	<i>Company E</i> Molecular Bio EP B 560556	04/15/1996 04/15/2010 License
Package Name	Description	Package Type
Influenza Sanitary Mask Know How		Standard
Asset	Asset Type	
USPTO 5597619	USPTO Document Technical Data	
License Agreements		
Agreement ID	Name	Effective Date Expiration Date Agreement Type
LA145433558	<i>Company E</i> Molecular Bio Flu Mask Know How	04/15/1996 04/15/2010 License
Package Name	Description	Package Type
Roche Molecular Logo		Standard
Asset	Asset Type	
USPTO TR 12345	Trademark	
License Agreements		
Agreement ID	Name	Effective Date Expiration Date Agreement Type
LA145433560	<i>Company E</i> Molecular Bio Roche Molecular Logo	04/15/1996 04/15/2010 License

FIG. 2988  
114B

142/193

Payment Allocation Report						
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type		
06/10/1998	LA145433556 Molecular-Bio	04/15/1996	04/15/2010	License		
Payment ID 1	Invoice Number 0101/1998	Payment Amount \$20,000.00	License Allocation Amt \$20,000.00			
	Package Name US 5597619					
	Asset USPTO 5597619					
		Allocation Percent 100%		Asset Allocation Amt \$20,000.00		
Payment ID 9	Invoice Number 0201/1998	Payment Amount \$1,000.00	License Allocation Amt \$1,000.00			
	Package Name US 5597619					
	Asset USPTO 5597619					
		Allocation Percent 100%		Asset Allocation Amount \$1,000.00		
Payment ID 10	Invoice Number 0301/1998	Payment Amount \$2,000.00	License Allocation Amt \$2,000.00			
	Package Name US 5597619					
	Asset USPTO 5597619					
		Allocation Percent 100%		Asset Allocation Amount \$2,000.00		
Agreement Total						\$23,000.00

143/193

Payment Allocation Report						
06/10/1998	Agreement ID	Name	Effective Date	Expiration Date	Agreement Type	
	LA145433557	Molecular-Bio EP B 560556	04/15/1996	04/15/2010	License	
	Payment ID	Invoice Number	Receipt Date	Payment Amount	License Allocation Amt	
	2	EP B 560556	06/10/1998	\$15,000.00	\$15,000.00	
		Package Name	Description			
		EP B 560556				
		Asset	Allocation Percent			
		EPO B 560556	100%			
			Asset Allocation Amt			
			\$16,000.00			
	Payment ID	Invoice Number	Receipt Date	Payment Amount	License Allocation Amt	
	11	EP B 560556	02/01/1998	\$1,000.00	\$1,000.00	
		Package Name	Description			
		EP B 560556				
		Asset	Allocation Percent			
		EPO B 560556	100%			
			Asset Allocation Amount			
			\$1,000.00			
	Payment ID	Invoice Number	Receipt Date	Payment Amount	License Allocation Amt	
	12	EP B 560556	05/01/1998	\$2,000.00	\$2,000.00	
		Package Name	Description			
		EP B 560556				
		Asset	Allocation Percent			
		EPO B 560556	100%			
			Asset Allocation Amount			
			\$2,000.00			
	Agreement Total					
						\$18,000.00

## Payment Allocation Report

Agreement ID	Name	Company A	Effective Date	Expiration Date	Agreement Type
08/10/1998	LA145433558	Molecular-Bio-Flu Mask Know How	04/15/1996	04/15/2010	License
Payment ID 3	Invoice Number	Receipt Date	Payment Amount	License Allocation Amt	
		03/20/1998	\$28,000.00	\$28,000.00	
	Package Name Description				
	Asset			Allocation Percent	Asset Allocation Amount
	USPTO 5597619			75%	\$21,000.00
				25%	\$7,000.00
Payment ID 13	Invoice Number	Receipt Date	Payment Amount	License Allocation Amt	
		02/01/1998	\$1,000.00	\$1,000.00	
	Package Name Description				
	Asset			Allocation Percent	Asset Allocation Amount
	USPTO 5597619			75%	\$750.00
				25%	\$250.00
Payment ID 14	Invoice Number	Receipt Date	Payment Amount	License Allocation Amt	
		05/01/1995	\$2,000.00	\$2,000.00	
	Package Name Description				
	Asset			Allocation Percent	Asset Allocation Amount
	USPTO 5597619			75%	\$1,500.00
				25%	\$500.00
Agreement Total					\$31,000.00

Copyright © 1998 by SmartPatents, Inc. Confidential. Do not duplicate.

FI6. 2016 115C



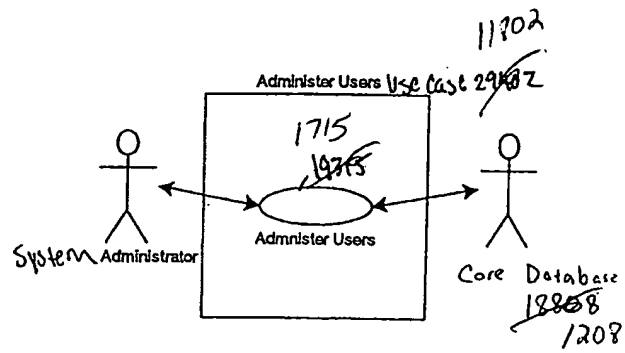
145/193

Royalty Statement Summary						
06/10/1998						
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type		
LA145433556	Molecular Bio USPTO 5597619	04/15/1996	04/15/2010	License		
Reporting Period	Invoice Number	Due Date	Receipt Date	Investigation Window		
2		01/01/1998	02/01/1998			
Line Number	Name	Units Sold	Revenue	Royalty Due		
1	Sanitary Mask	10	\$8,000.00	\$400.00		
2	Mask Filer	15	\$12,000.00	\$600.00		
			\$20,000.00	\$1,000.00		
Reporting Period	Invoice Number	Due Date	Receipt Date	Investigation Window		
3		04/01/1998	05/01/1998			
Line Number	Name	Units Sold	Revenue	Royalty Due		
1	Sanitary Mask	10	\$16,000.00	\$800.00		
2	Mask Filer	15	\$24,000.00	\$1,200.00		
			\$40,000.00	\$2,000.00		
Agreement Total			\$60,000.00	\$3,000.00		
Agreement ID	Name	Effective Date	Expiration Date	Agreement Type		
LA145433557	Molecular Bio EP B 560556	04/15/1996	04/15/2010	License		
Reporting Period	Invoice Number	Due Date	Receipt Date	Investigation Window		
4		01/01/1998	01/03/1998			
Line Number	Name	Units Sold	Revenue	Royalty Due		
1	Sanitary Mask	10	\$10,000.00	\$500.00		
2	Mask Filer	15	\$10,000.00	\$500.00		
			\$20,000.00	\$1,000.00		
Reporting Period	Invoice Number	Due Date	Receipt Date	Investigation Window		
5		04/01/1998	04/05/1998			
Line Number	Name	Units Sold	Revenue	Royalty Due		
1	Sanitary Mask	10	\$16,000.00	\$800.00		
2	Mask Filer	15	\$24,000.00	\$1,200.00		
			\$40,000.00	\$2,000.00		
Agreement Total			\$60,000.00	\$3,000.00		
Copyright © 1998 by SmartPatents, Inc. SmartPatents, Inc. Confidential. Do not duplicate.						

Fig. 292  
116

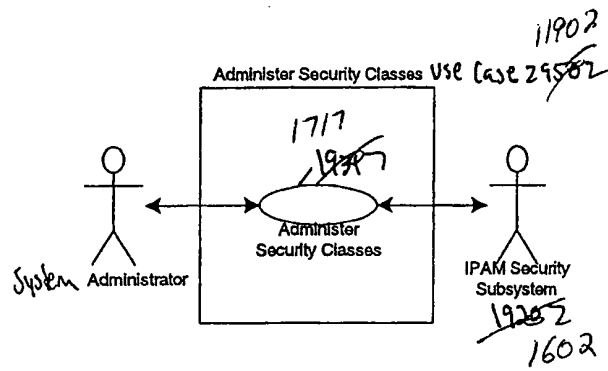
146/193

IP Asset - Historical Royalties									
Package ID	Package Name	Description							
2	US 5597619								
Asset Name	USPTO 5597619								
Agreement ID	LA145433556	Name	Effective Date	Expiration Date	Allocation Percent				
					100%				
					Agreement Type				
LA145433556	Molecular Bio USPTO 5597619	Receipt Date	License Allocation	Asset Allocation	License	Running Total			
		01/01/1998	\$20,000.00	\$20,000.00		\$20,000.00			
		02/01/1998	\$1,000.00	\$1,000.00		\$21,000.00			
		05/01/1998	\$2,000.00	\$2,000.00		\$23,000.00			
			\$23,000.00	\$23,000.00					
LA145433564	BioSpecifics USPTO 5597619	Receipt Date	License Allocation	Asset Allocation	License	Running Total			
		05/15/1998	\$5,000.00	\$5,000.00		\$5,000.00			
		02/01/1998	\$500.00	\$500.00		\$5,500.00			
		05/01/1998	\$2,000.00	\$2,000.00		\$7,500.00			
			\$7,500.00	\$7,500.00					
LA145433565	Amgen USPTO 5597619	Receipt Date	License Allocation	Asset Allocation	License	Running Total			
		02/01/1998	\$1,000.00	\$1,000.00		\$1,000.00			
		05/01/1998	\$2,000.00	\$2,000.00		\$3,000.00			
			\$3,000.00	\$3,000.00					
			\$33,500.00	\$33,500.00					



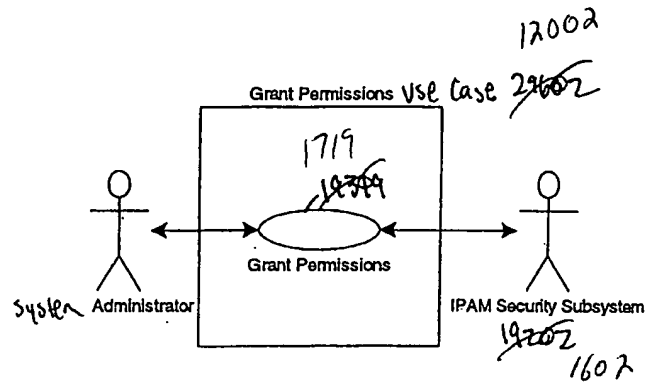
118  
FIG. 294

148/193



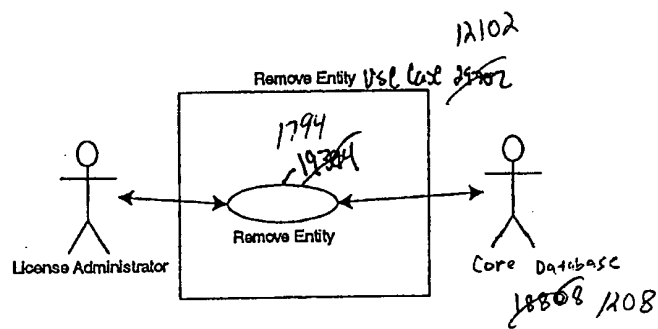
119  
FIG. 29582

149/193

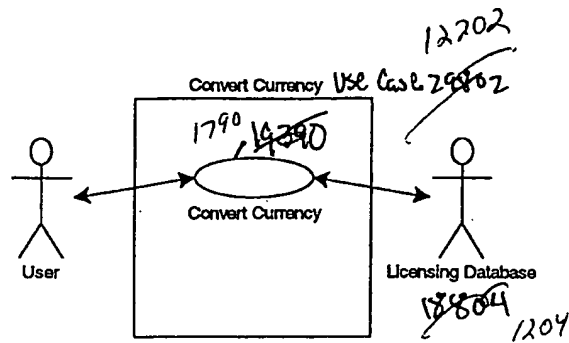


120  
FIG. 7

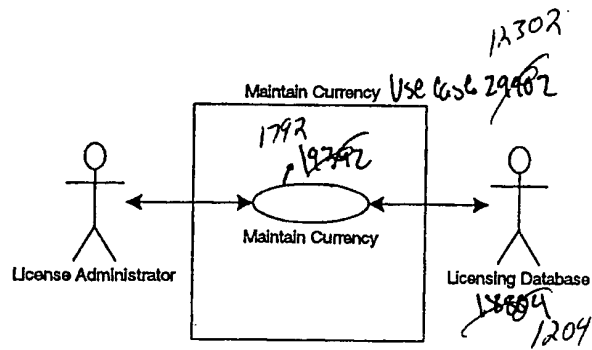
150 / 193



121  
FIG. 287

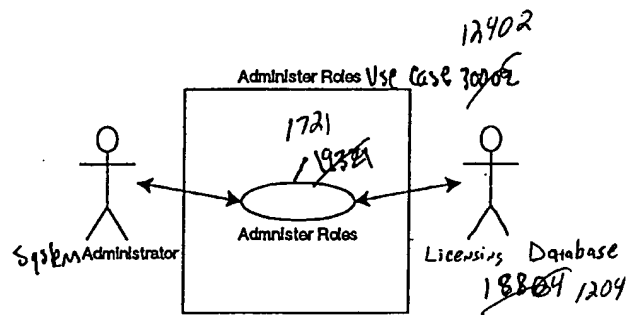


122  
FIG. 298



123  
Fig. 249





124  
FIG. 300

154/193

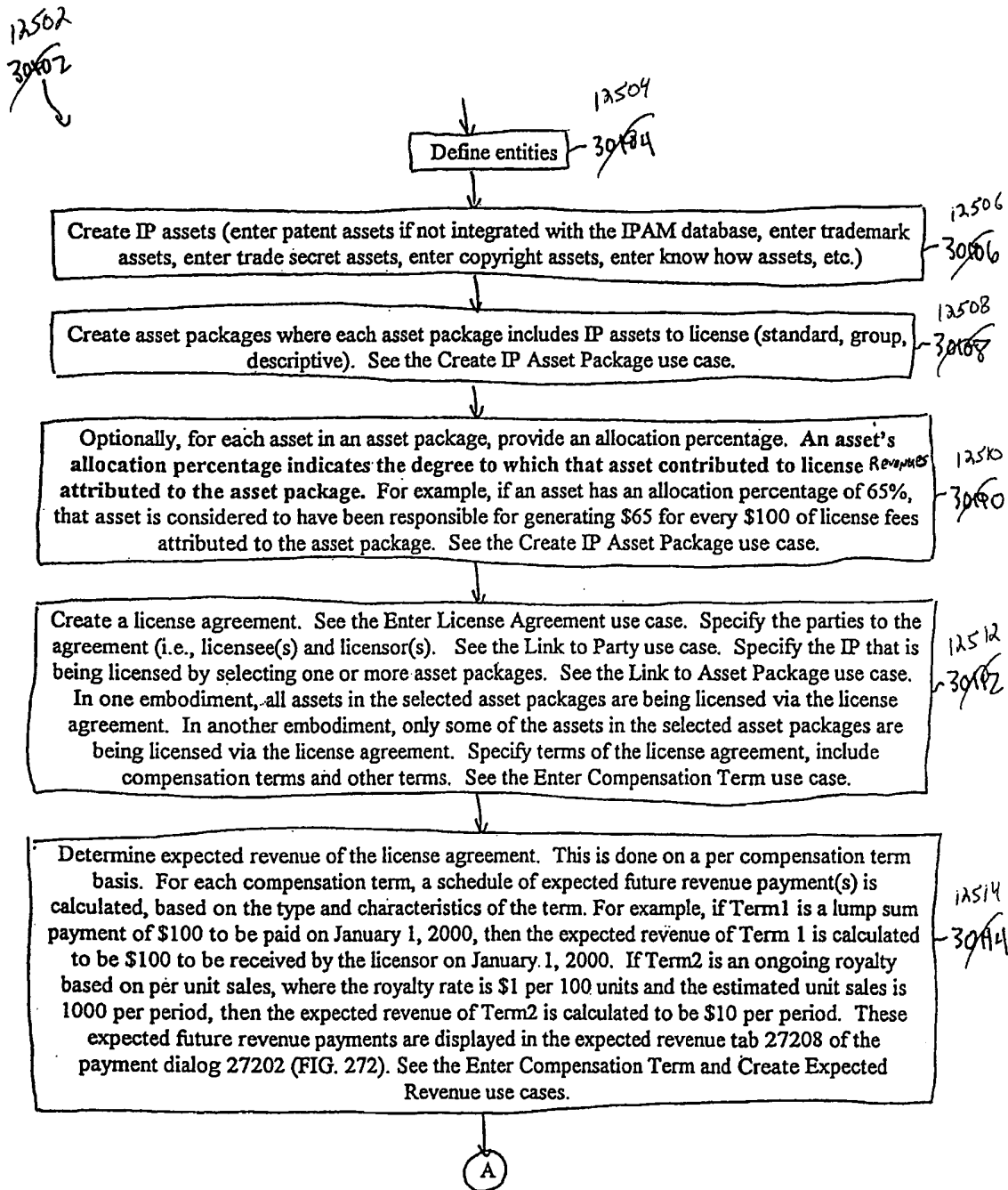
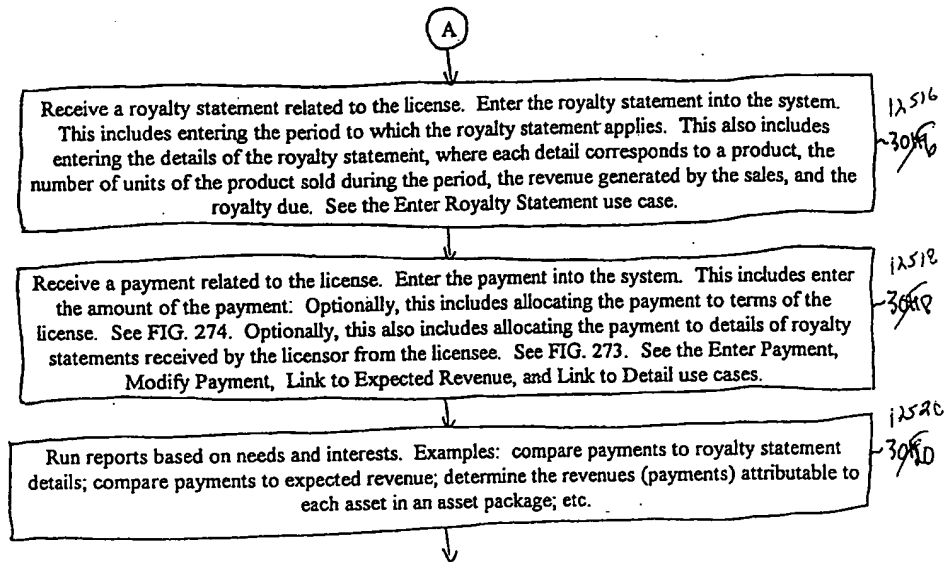


FIG. 301A

125A

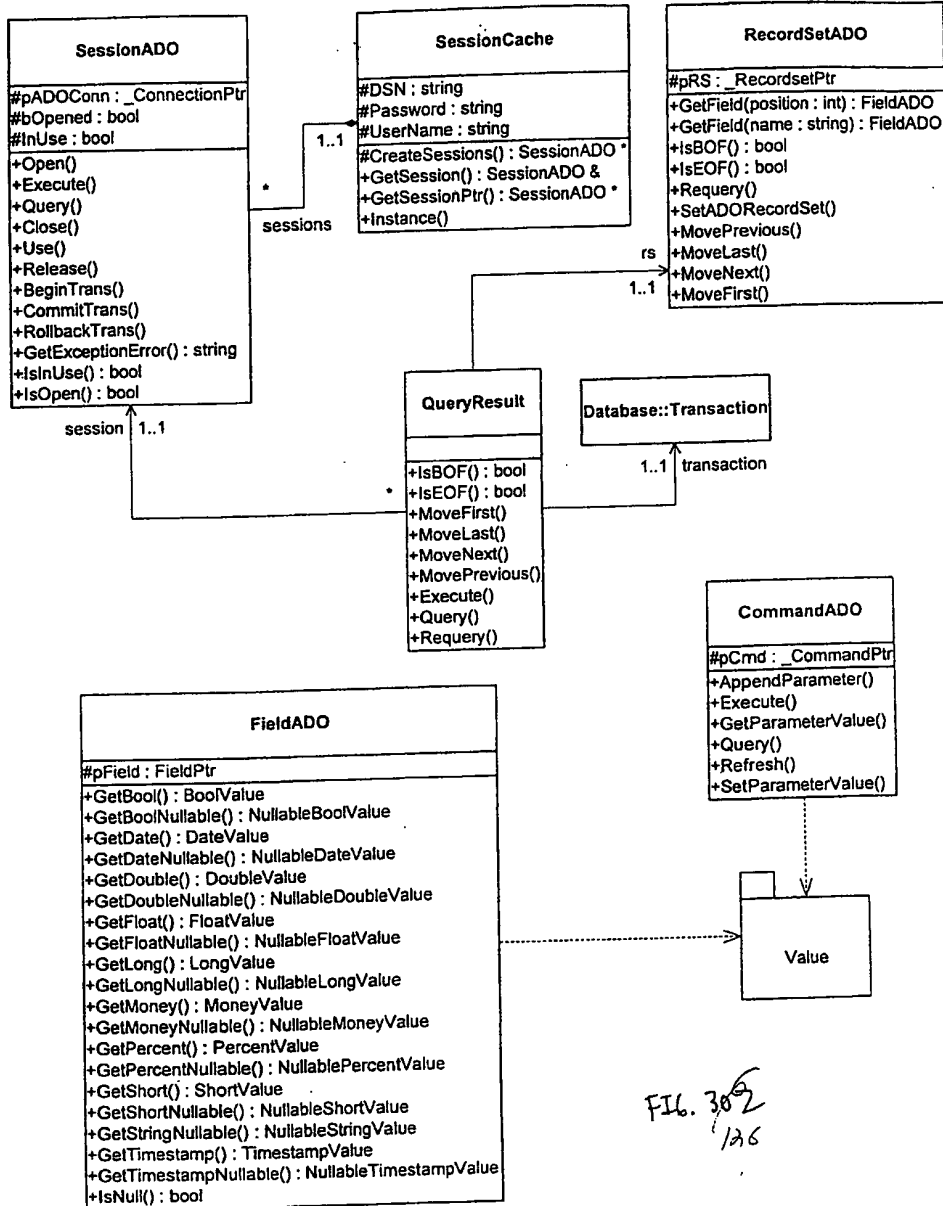
155/193

FIG. 307B  
135B

156/193

## Logical View

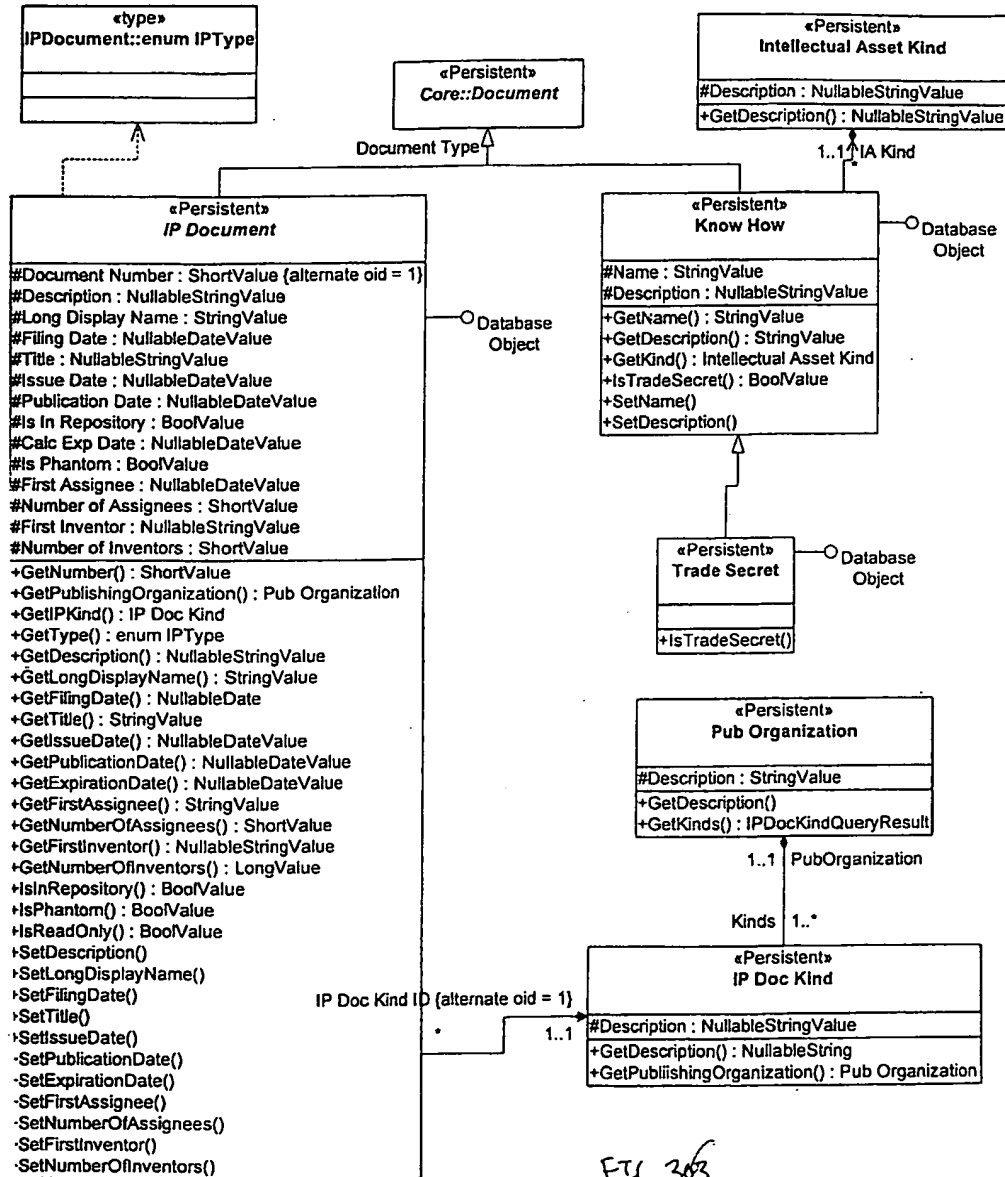
## ADO Package

FIG. 306  
126

157/193

Logical View

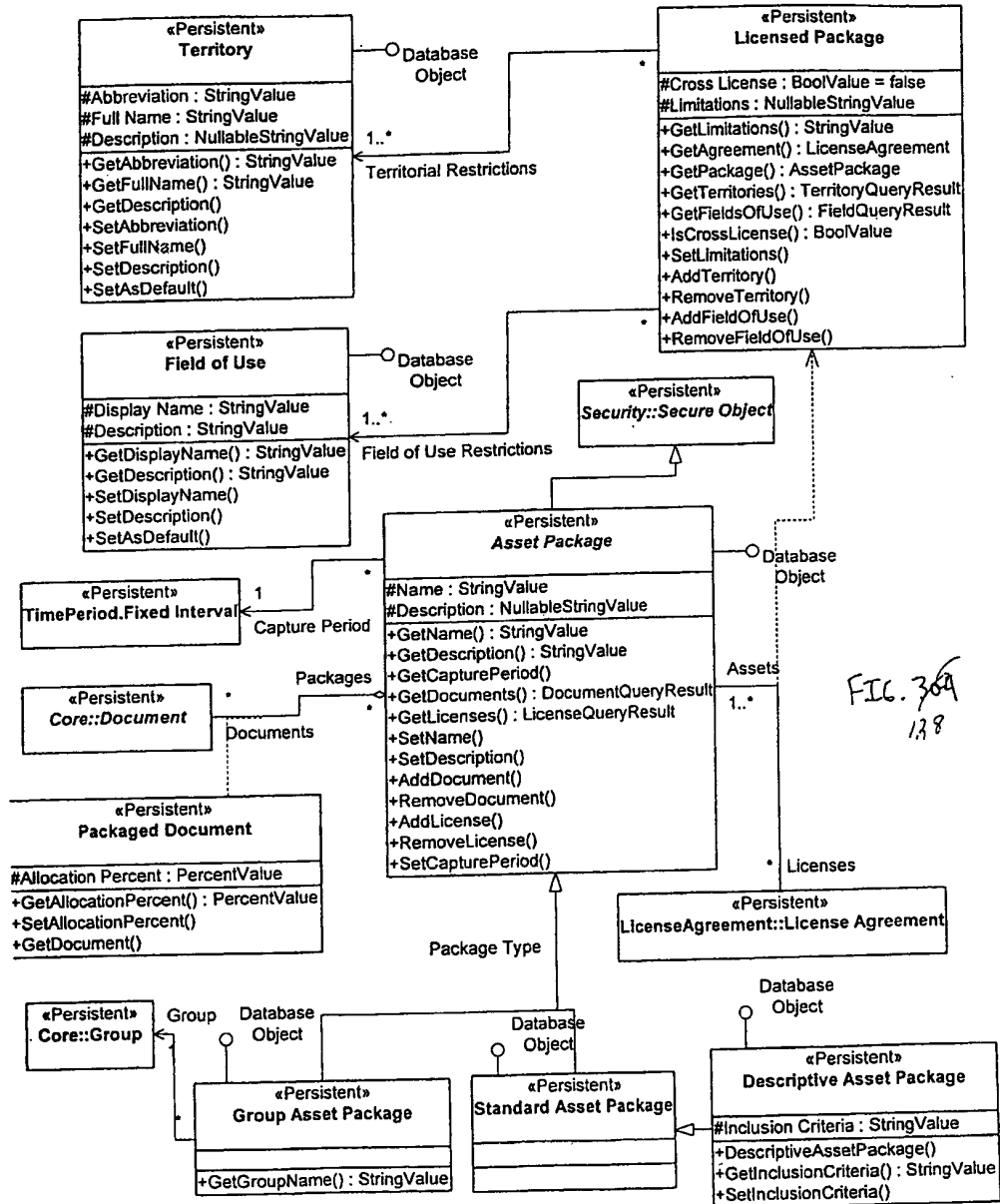
## IP Asset Package

FIG. 303  
137

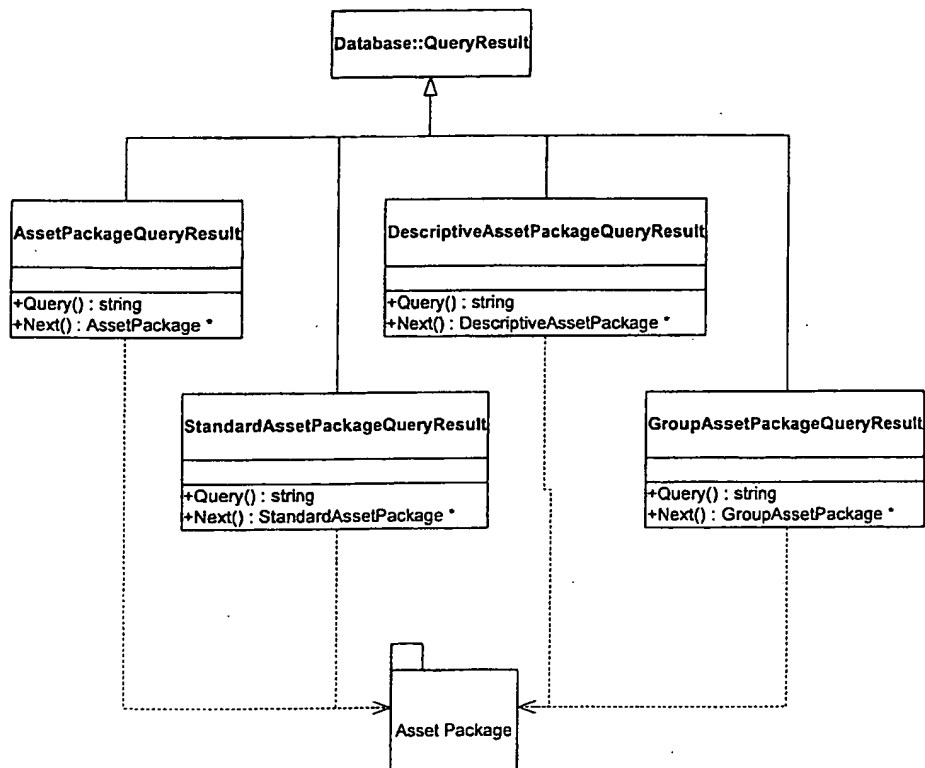
158/193

### Logical View

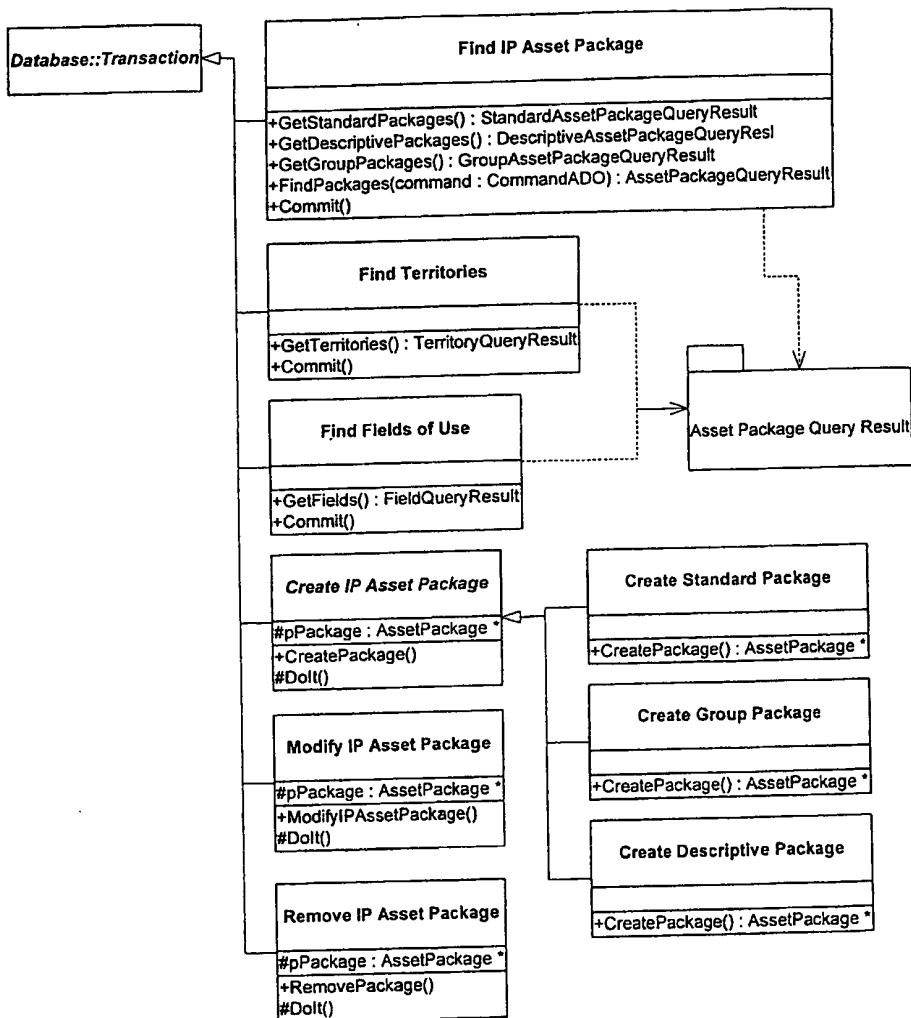
## Asset Package Package



Logical View

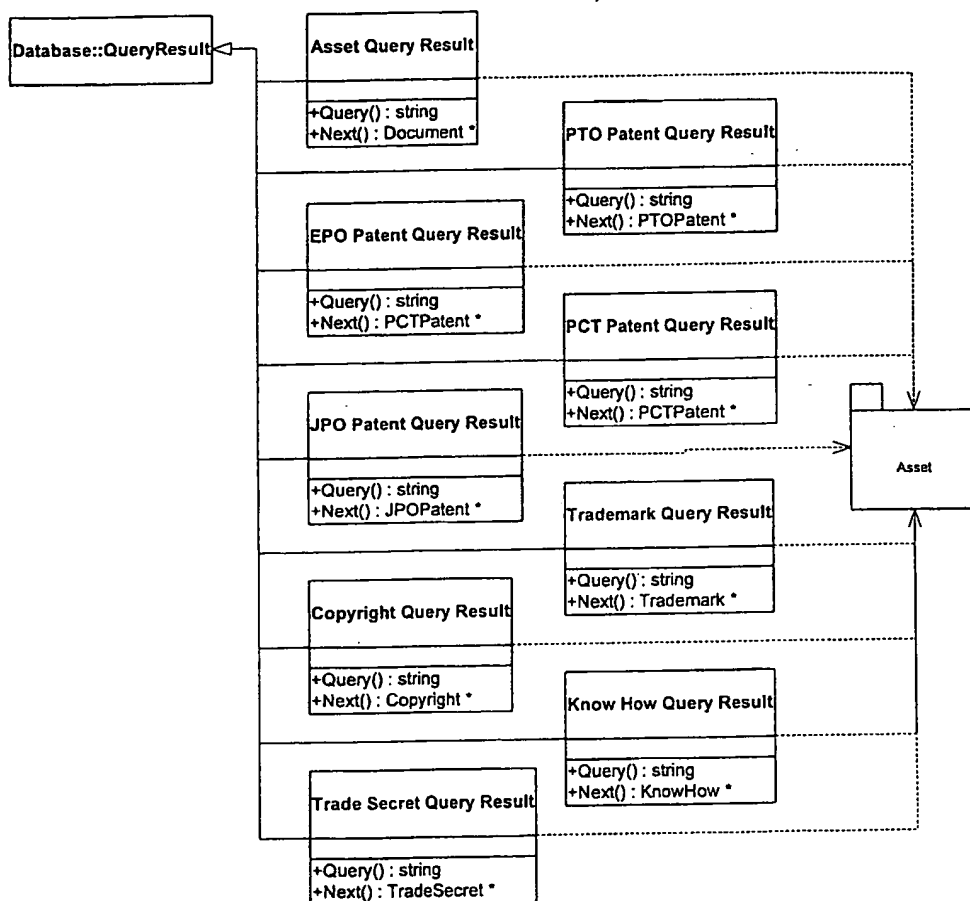
**Asset Package Query Result Package**FIG. 305  
129

Logical View

**Asset Package Transactions**FIG. 306  
130



.logical View

**Asset Query Result Package**FIG. 309  
131

162 / 193

Logical View

# Asset Transactions

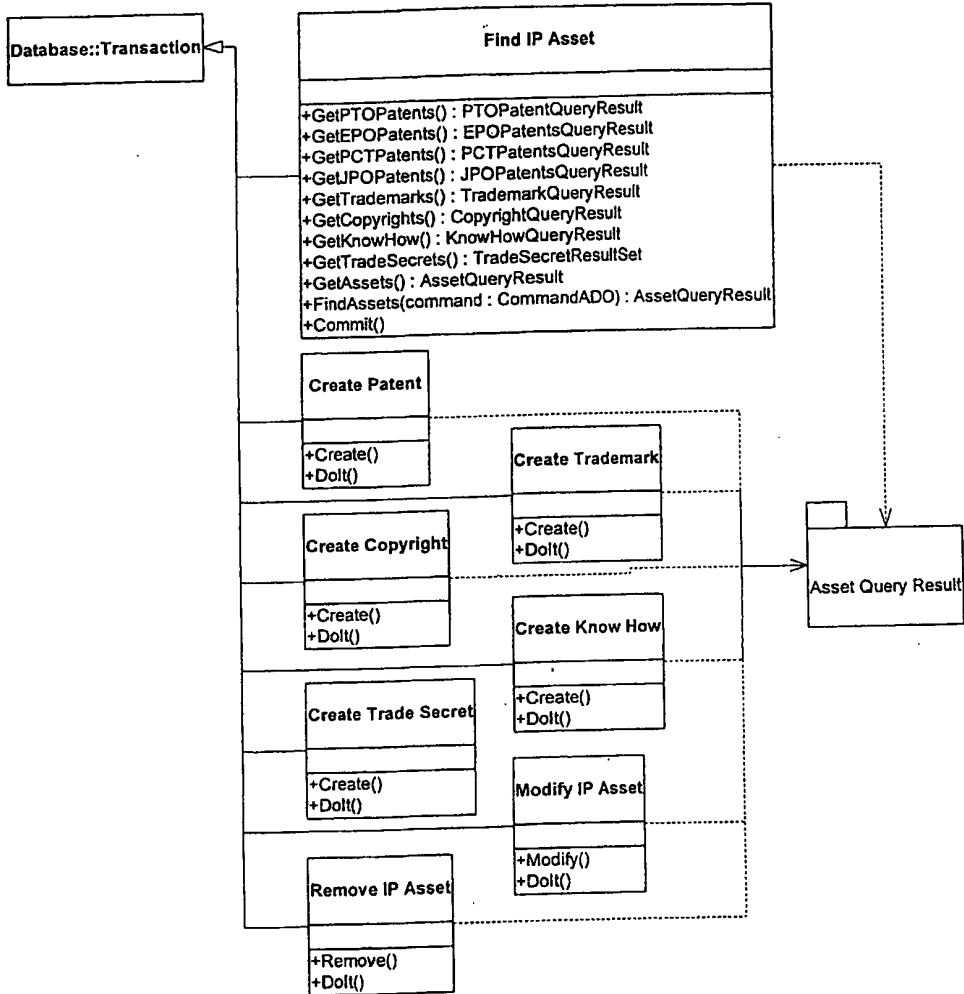
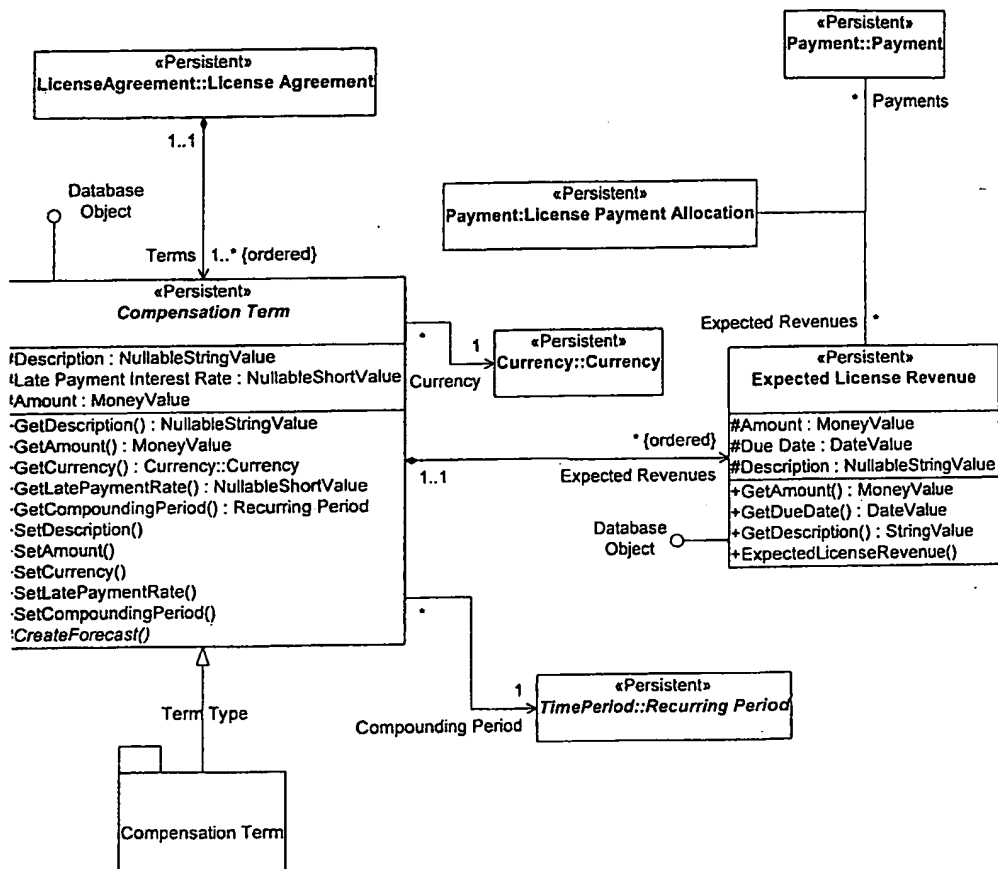


FIG. 308  
132

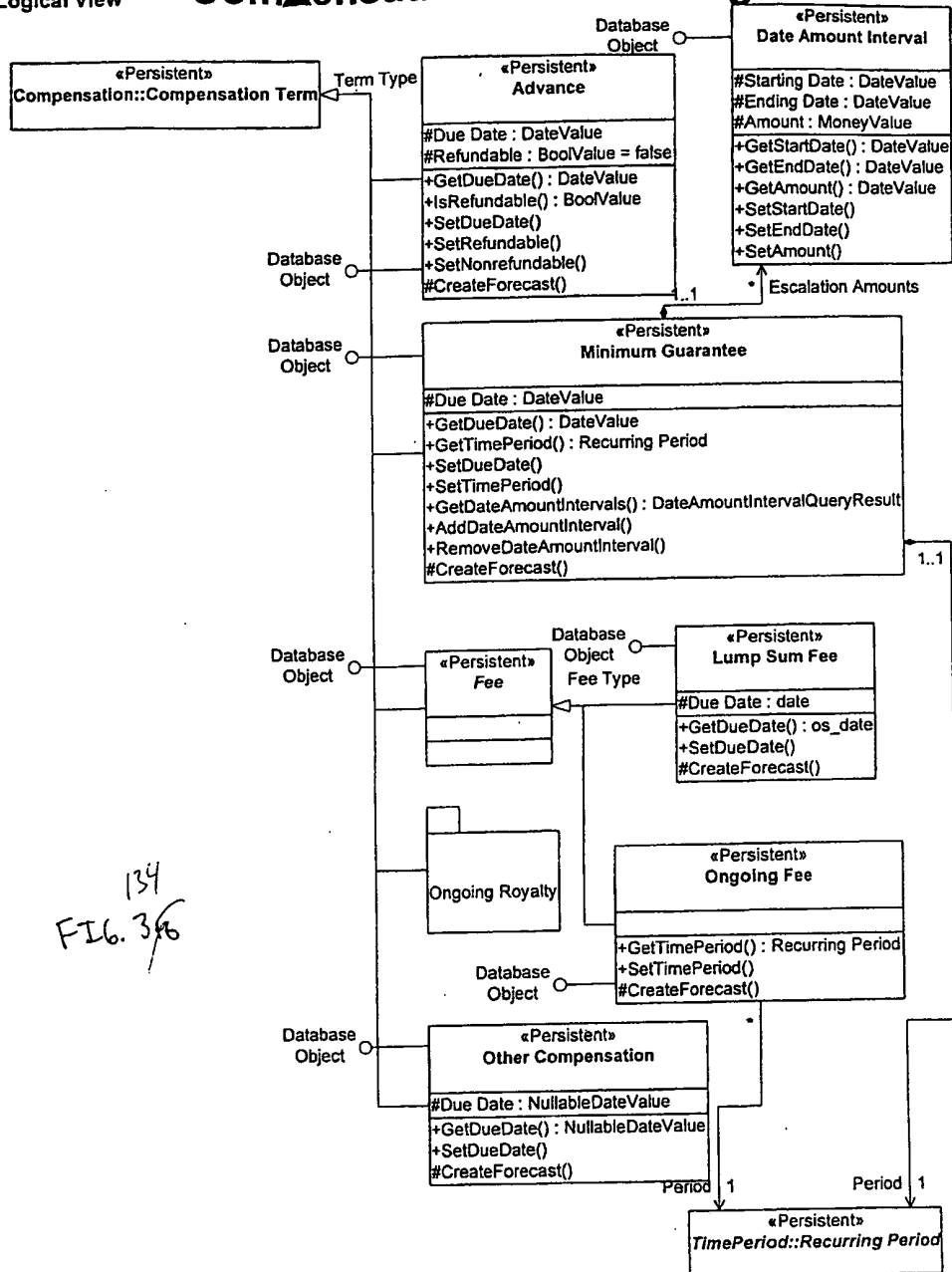
Logical View

## Compensation Package

FIG. 369  
/33

### Logical View

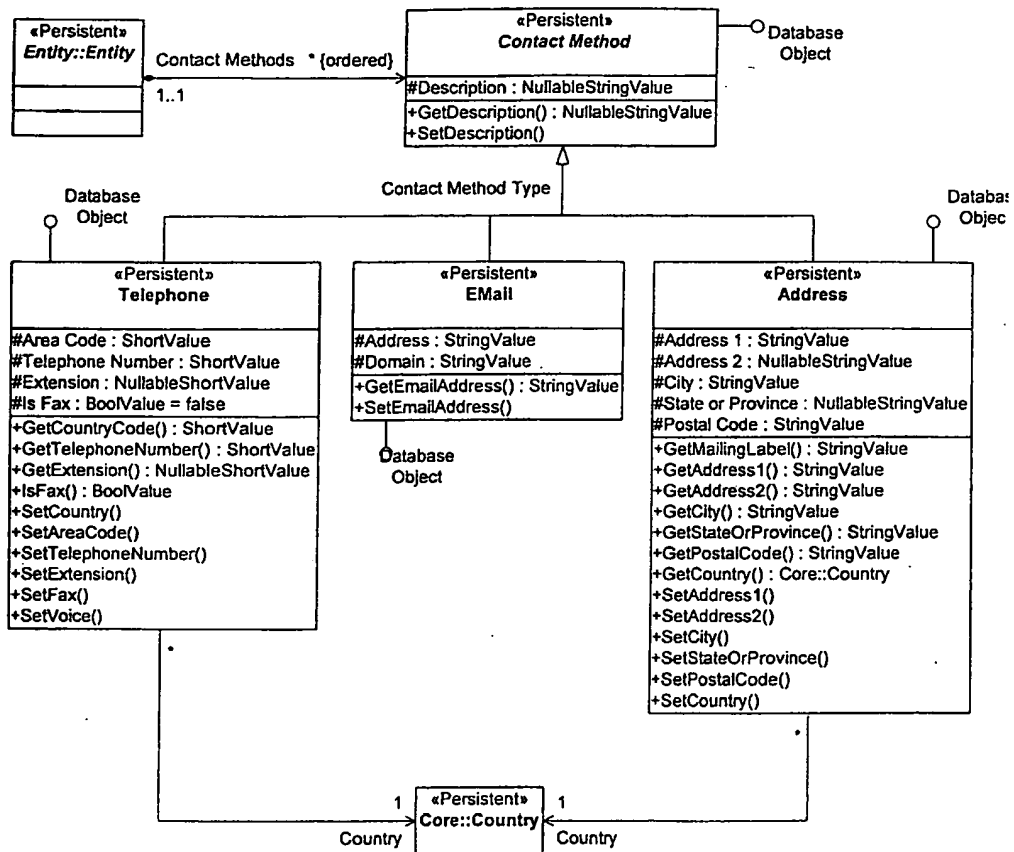
## Compensation Term Package



165/193

Logical View

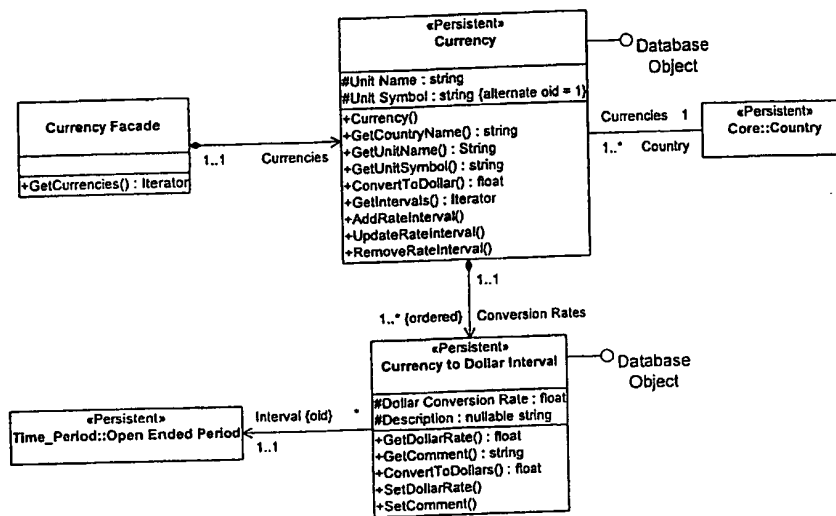
## Contact Package

135  
FIG. 3A

166/193

Logical View

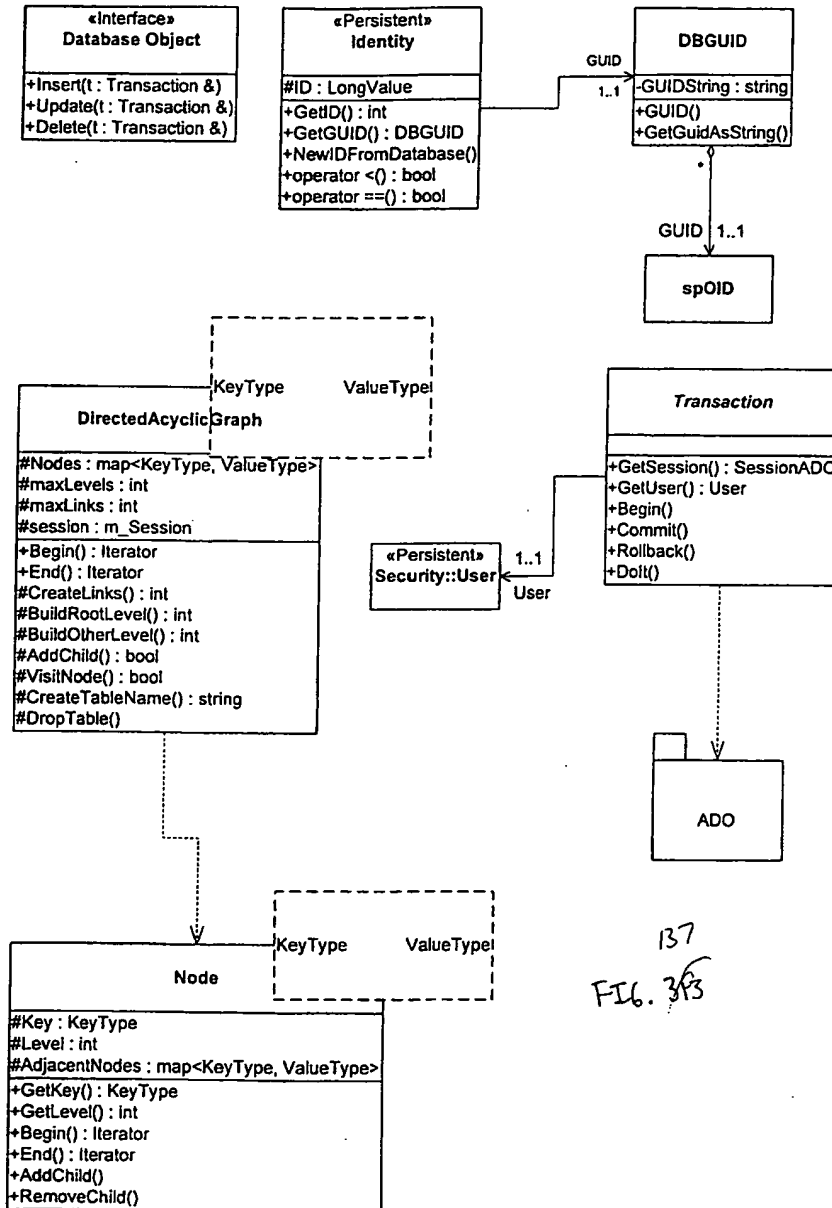
## Currency Package

136  
FIG. 2A

167/193

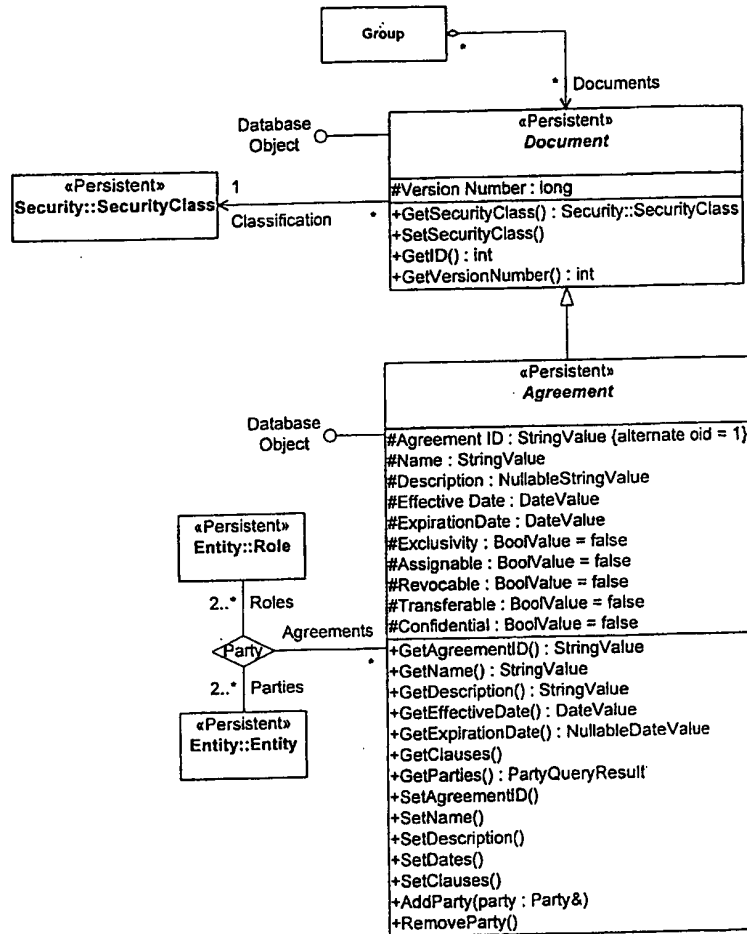
## Logical View

## Database Package

137  
FIG. 383

168 / 193

Logical View

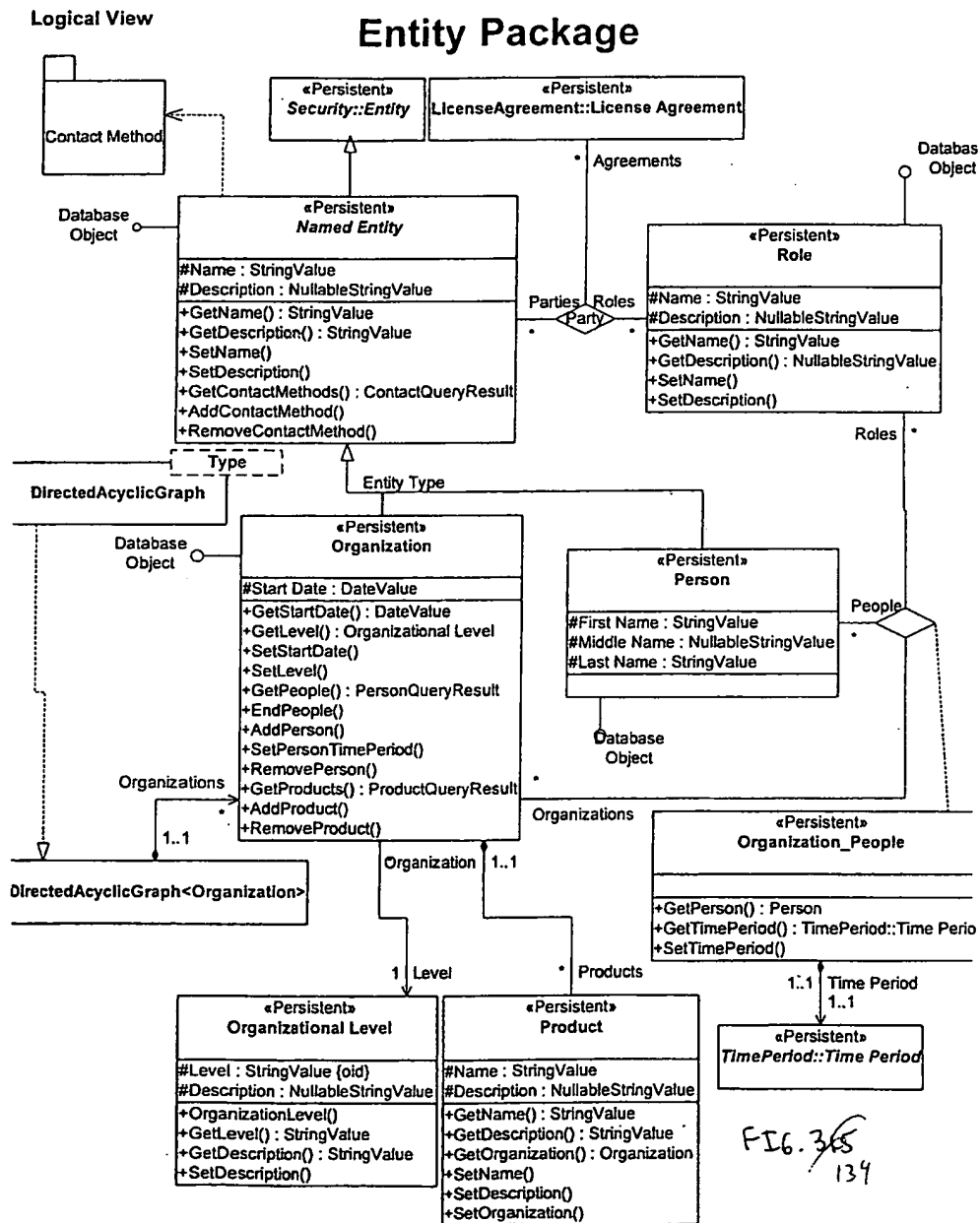
**Document Package**

139

FIG. 3A

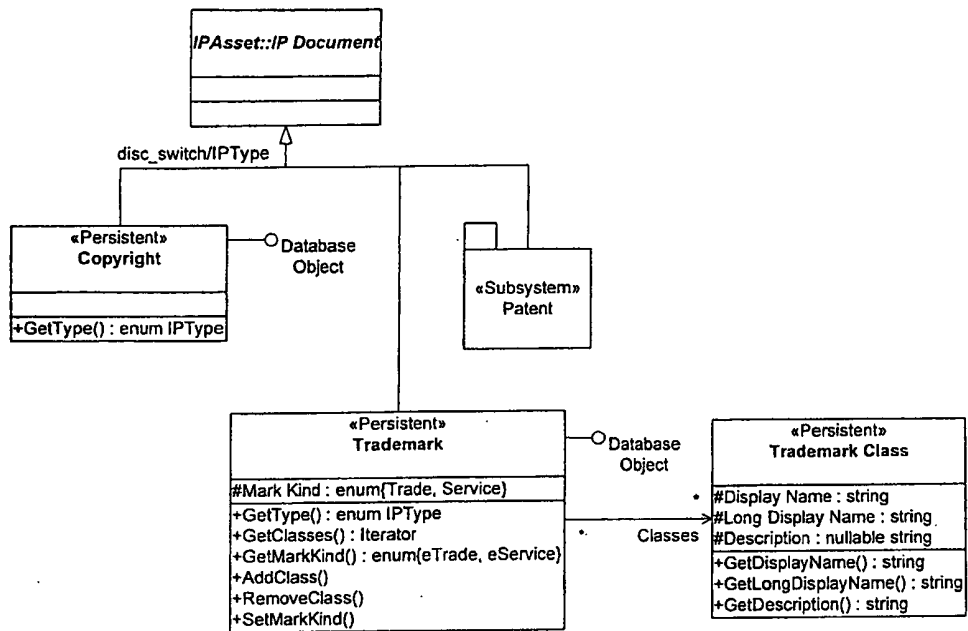


169/193



170/193

Logical View

**II Document Package**

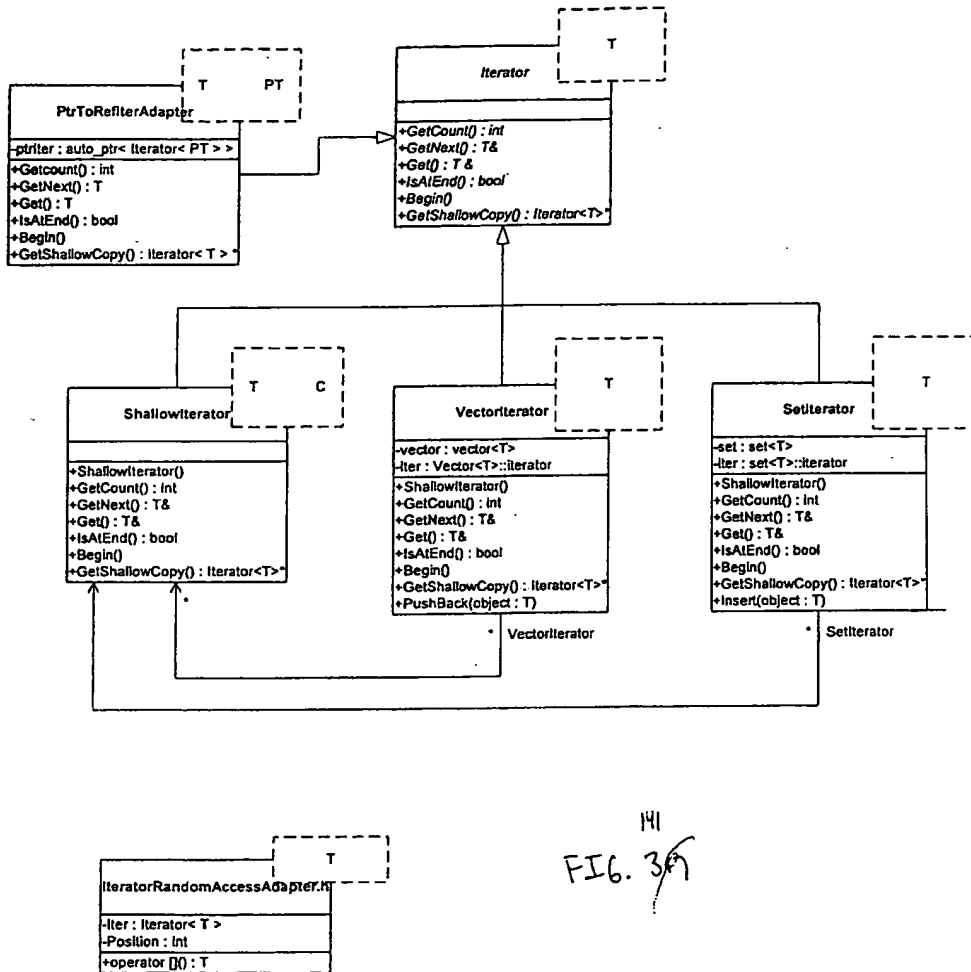
140

FIG. 346

171/193

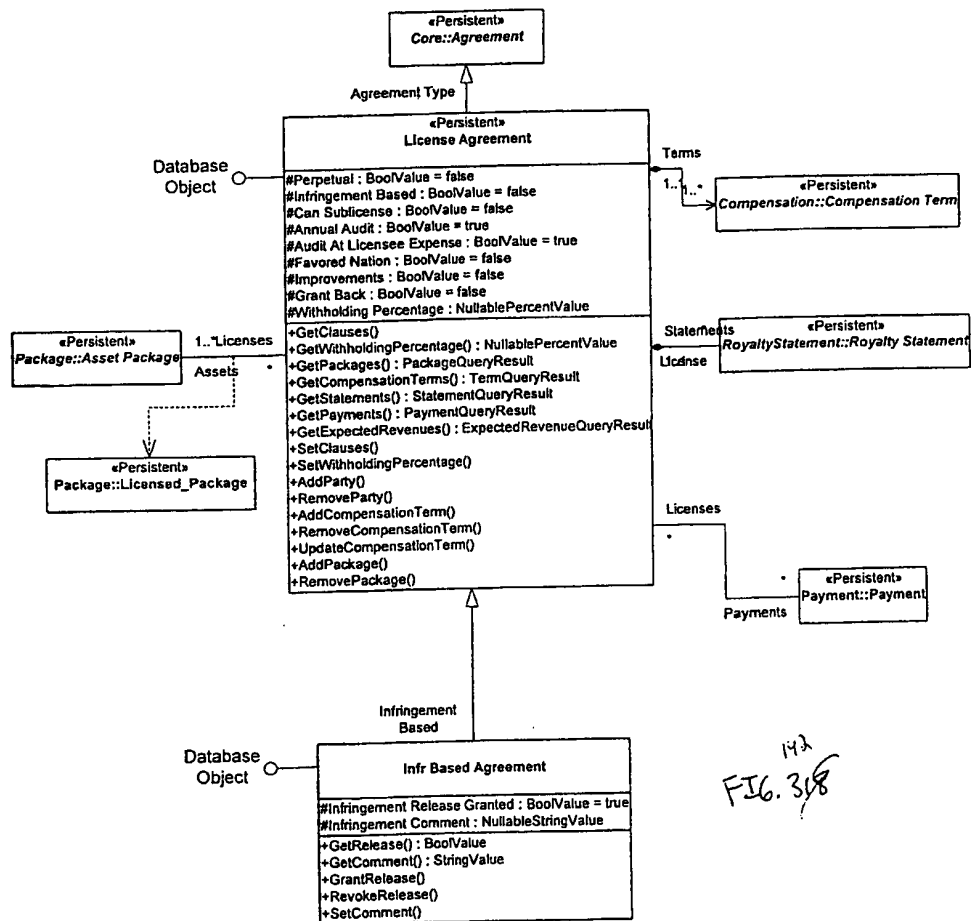
Logical View

## Iterator Package

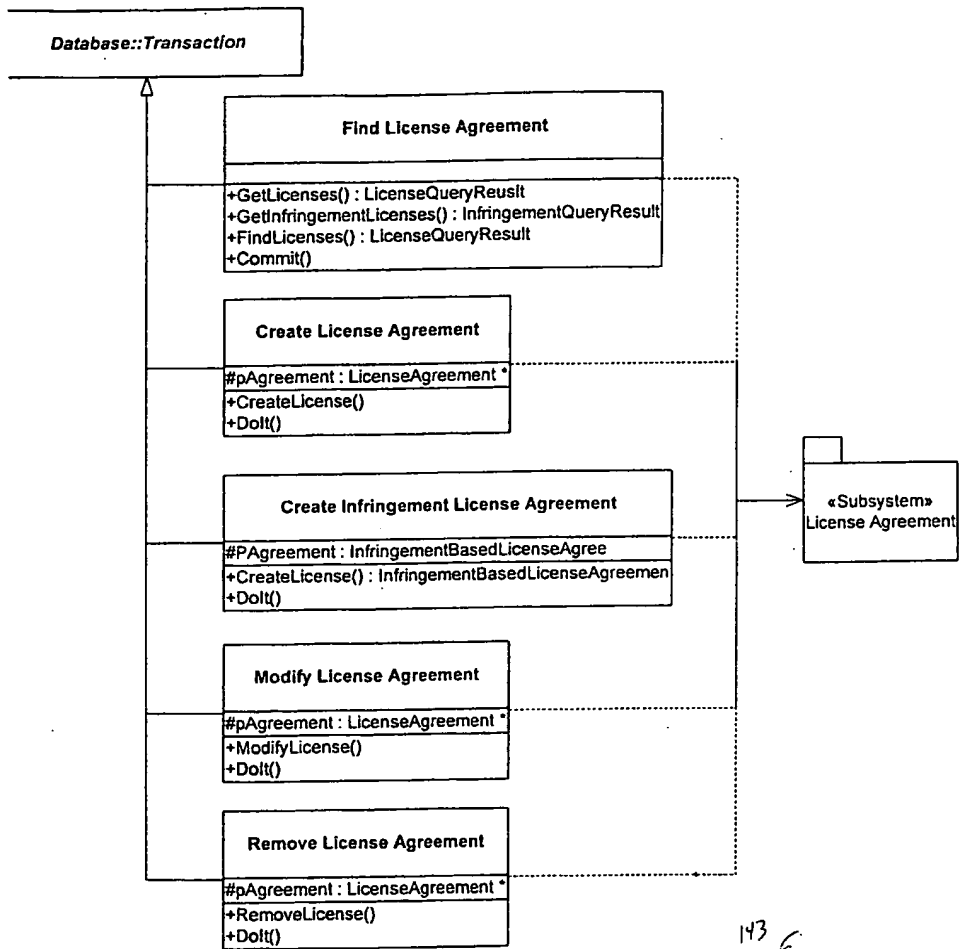
141  
FIG. 36

Logical View

## License Agreement Package



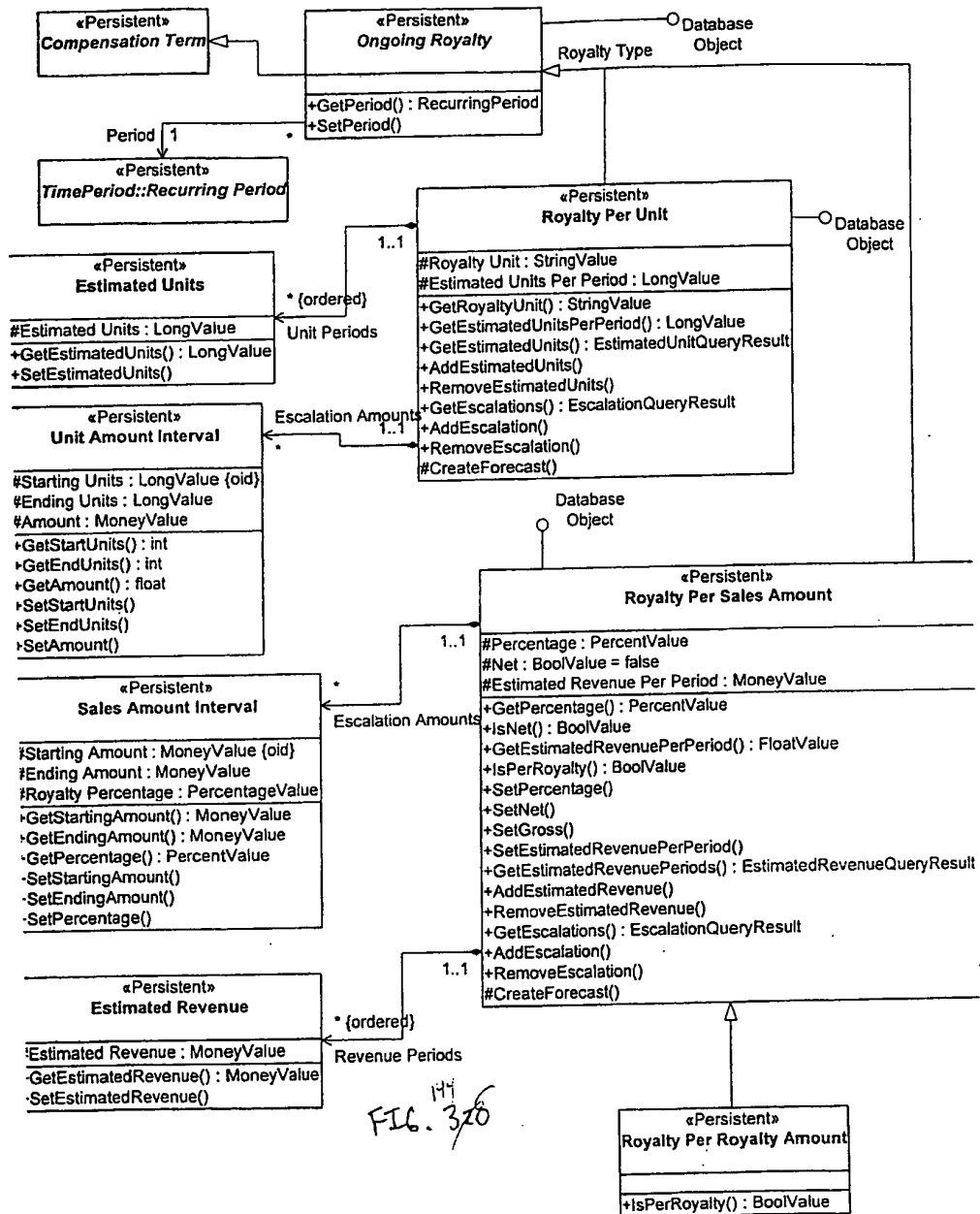
Logical View

**License Agreement Transaction Package**173  
FIG. 319

174/193

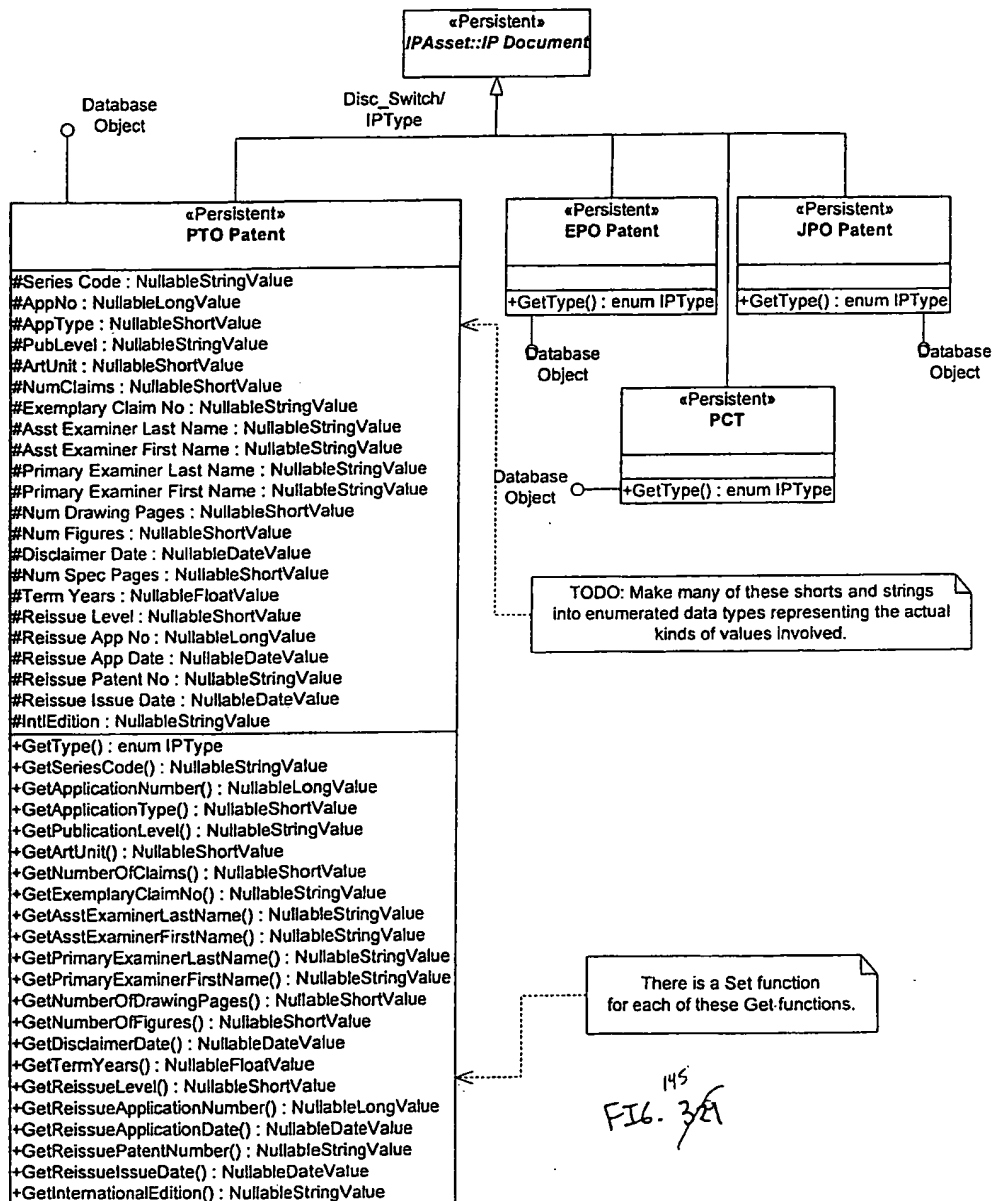
Logical View

## Ongoing Royalty Package



### Logical View

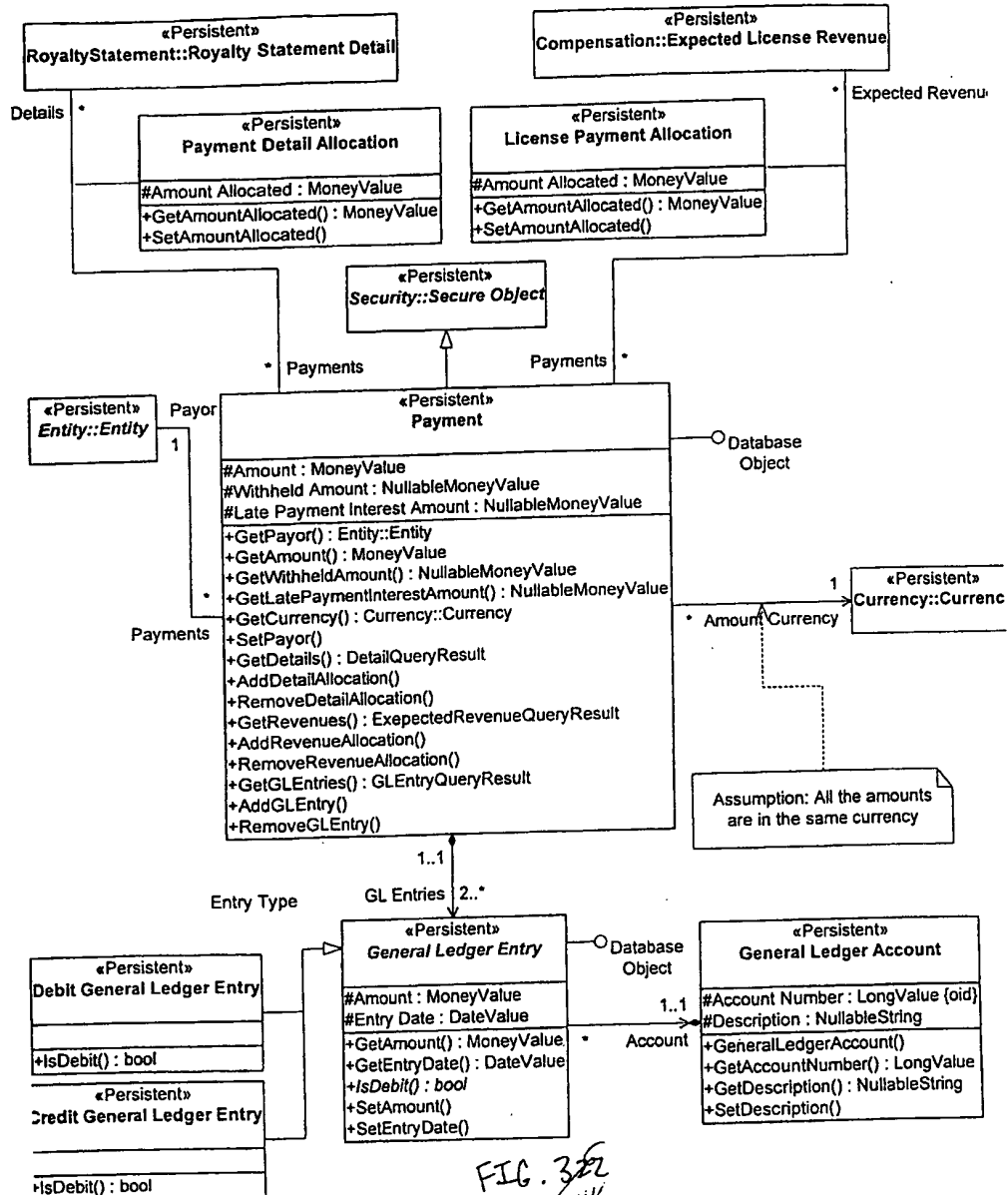
## Patent Package



176/193

Logical View

## Payment Package

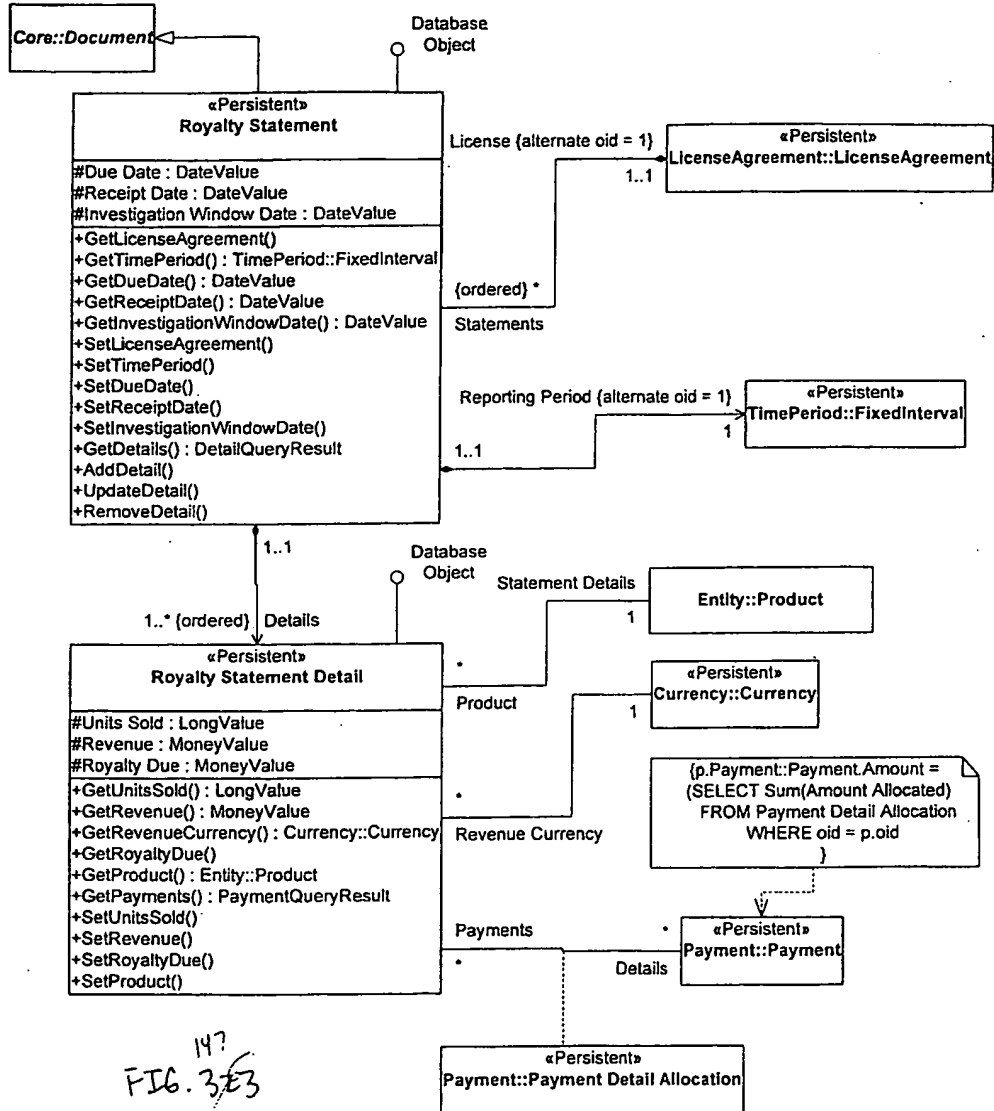




177/193

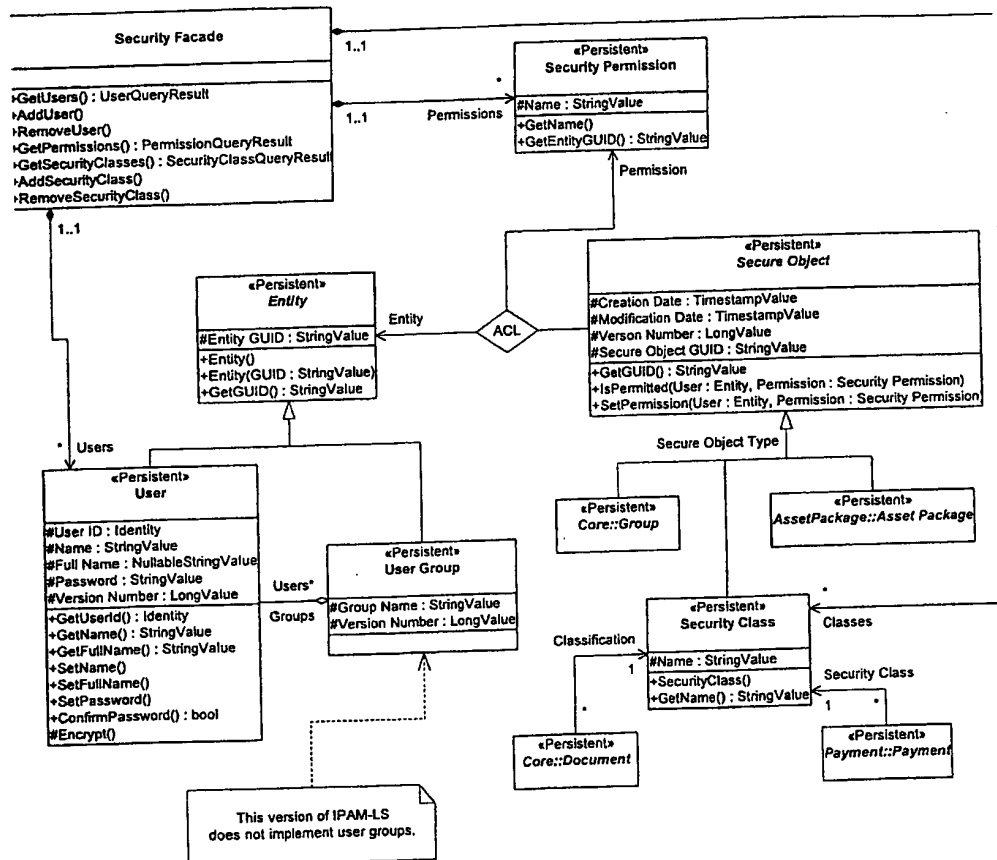
Logical View

## Royalty Statement Package



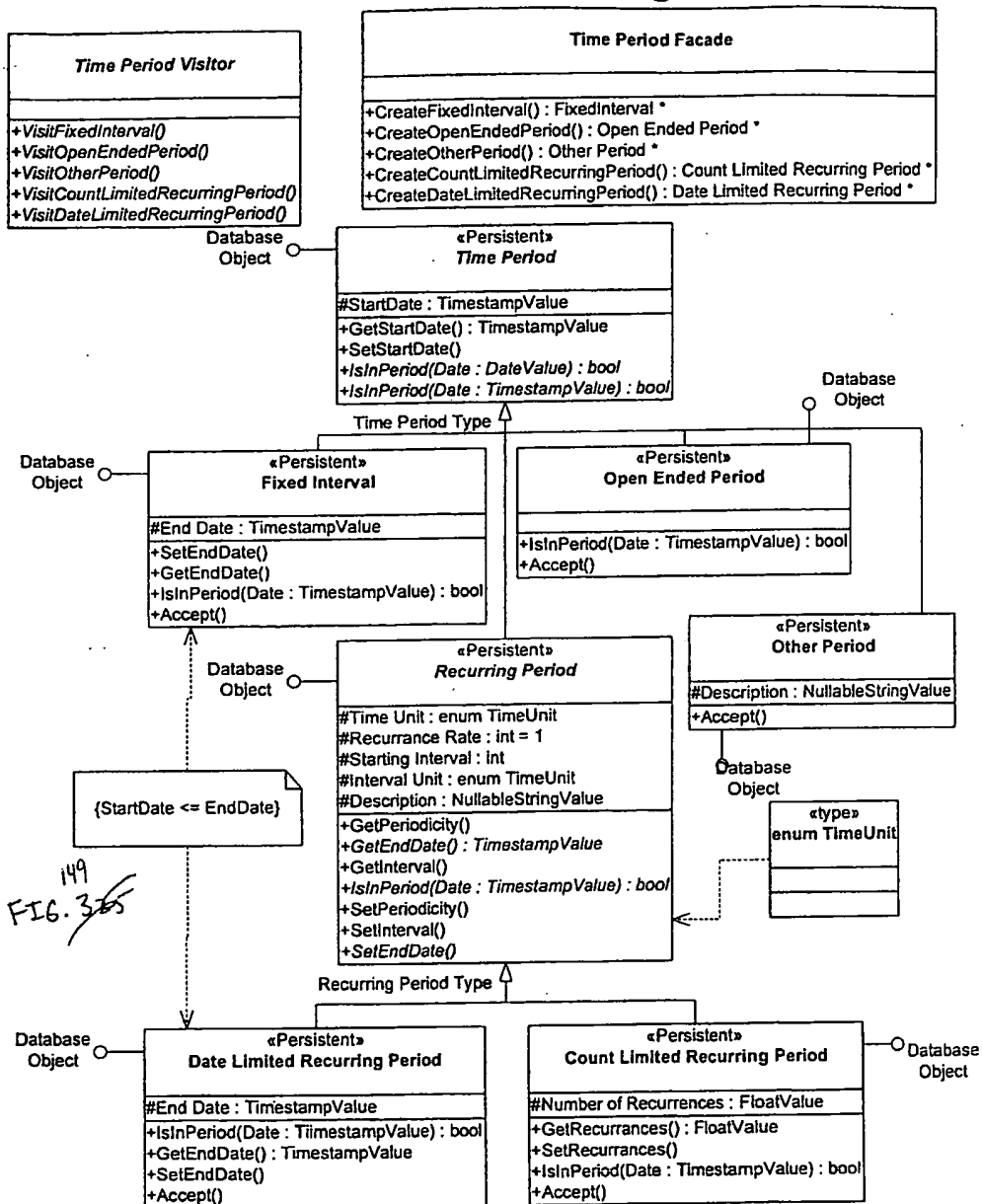
## Logical View

## Security Package

148  
FIG. 324

Logical View

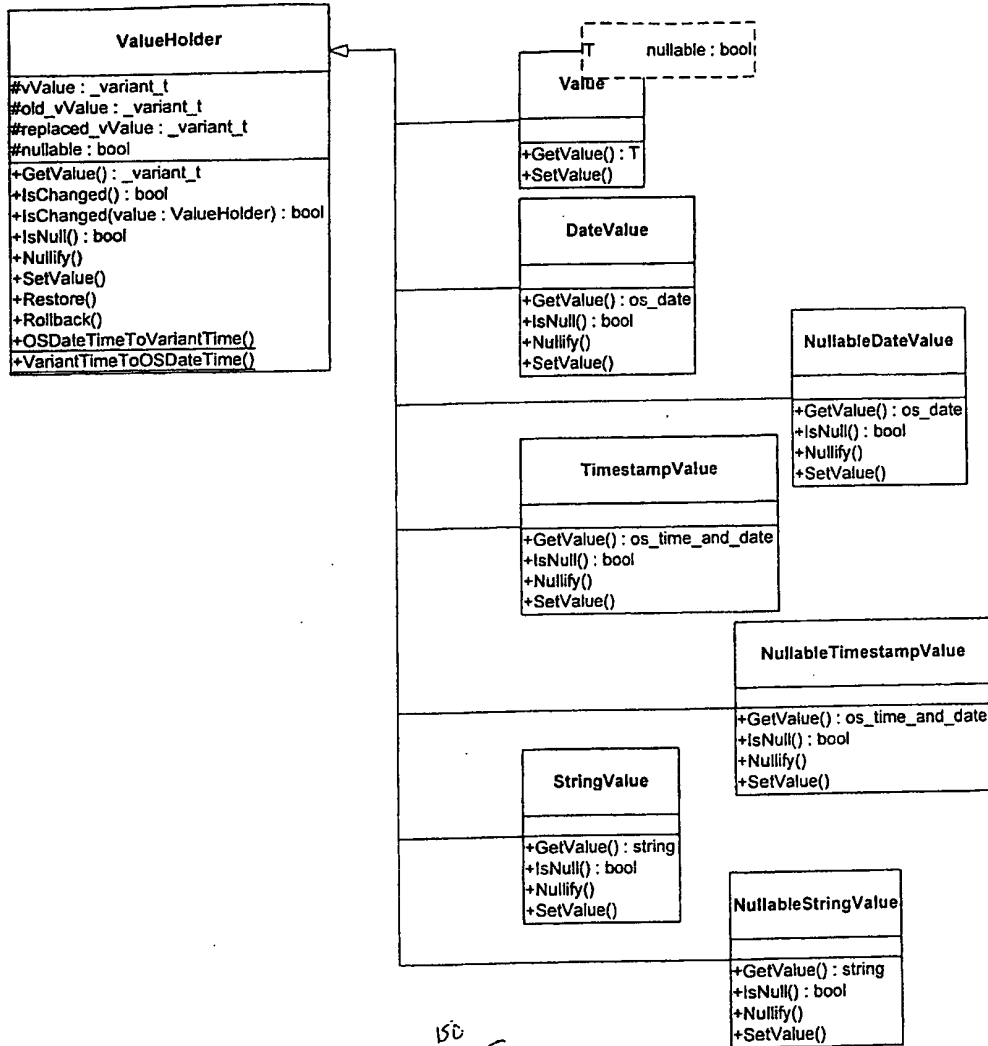
## Time Period Package

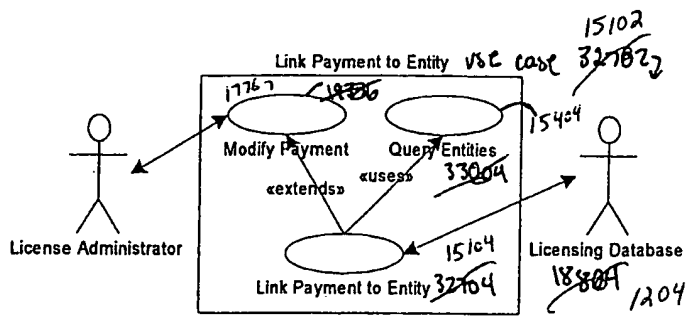


180/193

.logical View

## Value Package

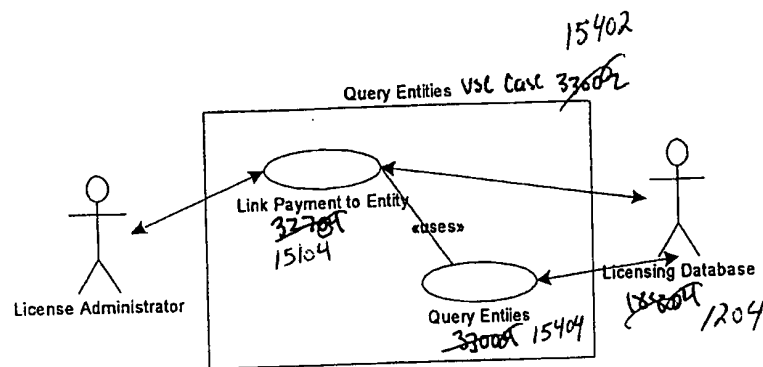
150  
FIG. 326



151  
FIG. 327



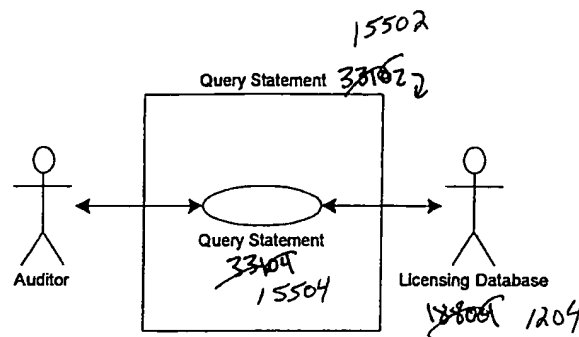




154  
FIG. 380

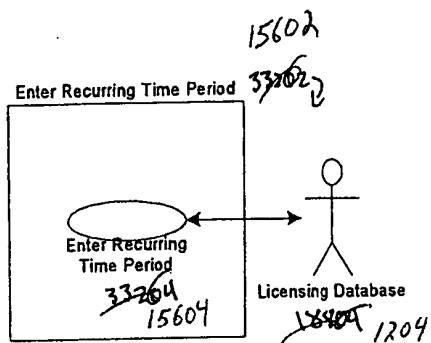


185/193



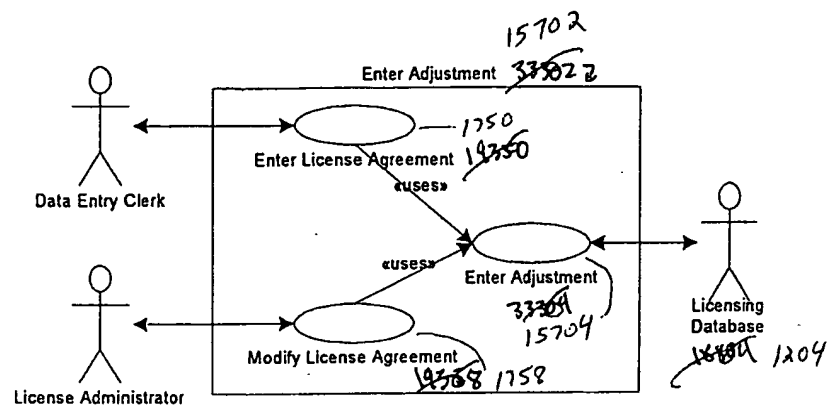
155  
FIG. 351

186/193



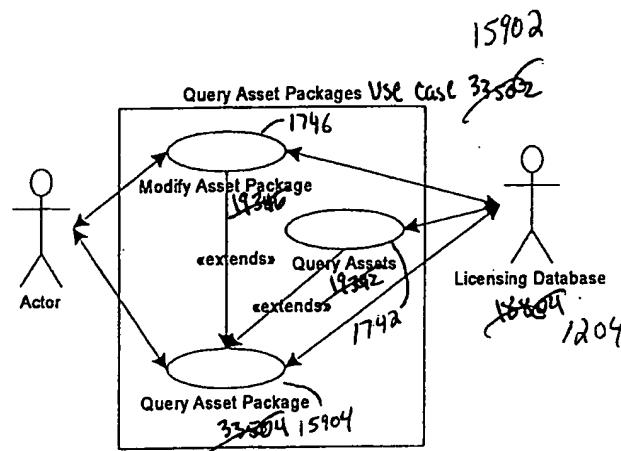
156  
FIG. 332

187/193



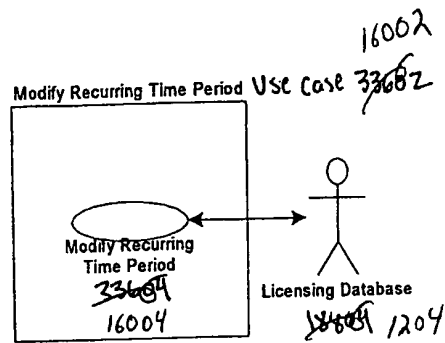
157  
FIG. 330





159  
FIG. 335

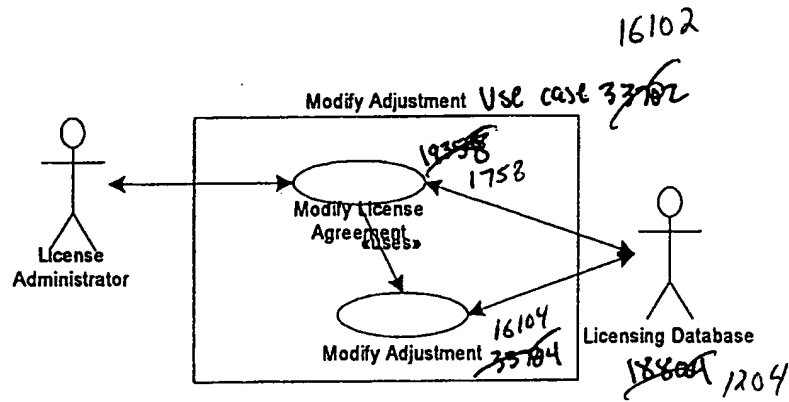
190/193



160

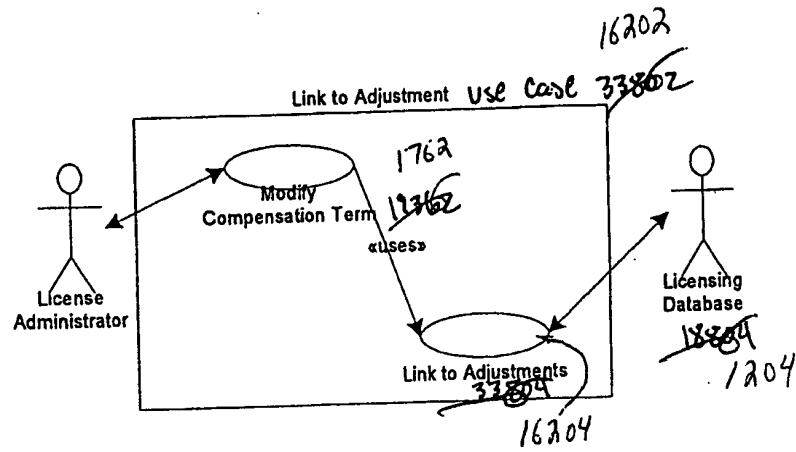
FIG. 386

191/193



161  
FIG. 331

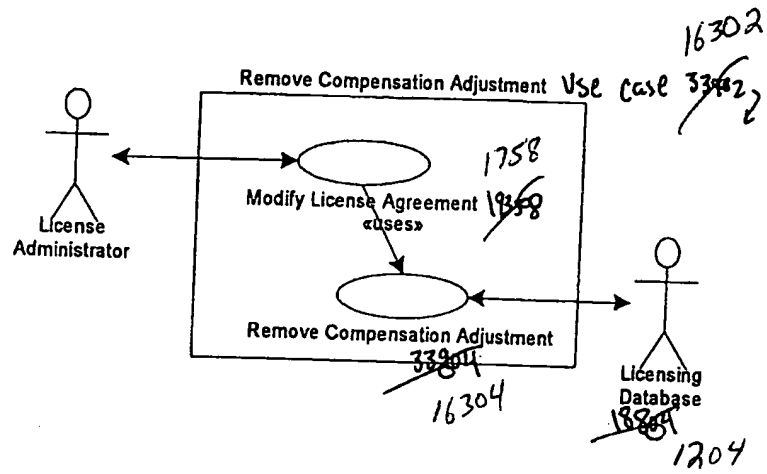
192/193



162  
FIG. 338



193 / 193



163  
FIG. 336

# INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US99/19050

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 17/30, 17/60  
US CL : 707/104; 705/35

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/104, 526; 705/28, 30, 35, 36

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, WEST, DIALOG  
search terms: intellectual property management, royalty, license, analysis, database management

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,892,900 A (GINTER et al) 06 April 1999	1-45
A,P	US 5,875,431 A (HECKMAN et al) 23 February 1999	1-45
A	US 5,634,012 A (STEFIK et al) 27 May 1997	1-45
A	US 5,530,520 A (CLEARWATER) 25 June 1996	1-45
A	US 5,444,779 A (DANIELE) 22 August 1995	1-45
A	US 5,251,294 A (ABELOW) 05 October 1993	1-45

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A* document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E* earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A* document member of the same patent family
O* document referring to an oral disclosure, use, exhibition or other means	
P* document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

08 NOVEMBER 1999

Date of mailing of the international search report

19 NOV 1999

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231  
Facsimile No. (703) 305-3230

Authorized officer

MICHAEL RAZAVI

Telephone No. (703) 305-3900

*James R. Matthews*